

MINISTERE DE
L'ENSEIGNEMENT SUPERIEUR
UNIVERSITE DE DOUALA



MINISTRY OF HIGHER
EDUCATION
UNIVERSITY OF DOUALA



ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE DE DOUALA

BP : 2701 DOUALA

TEL : +237 697 542 240

contact@enspd-udo.cm

Division de la scolarité et des Etudes

Service de la Formation par Alternance et de l'insertion Professionnelle

Document final portant sur le Système
d'Authentification des Diplômes Camerounais

Liste des membres du groupe :

AGASSEM SANDA FLORENTIN	24G01072
KWIDA NGOBO GRACE KEYS	24G01099
MAMOUDOU BAOURO	24G01105
NGO BIHOLONG MARTHE FRANCINE	24G01120
NZONNOU NJAMPA GRACE	22G00350

Enseignant : *Dr IHONO CK*

Année scolaire : 2025-2026

Table des matières

CHAPITRE 1 : ANALYSE ET CONCEPTION DE L'APPLICATION	3
SECTION 1 : CONCEPTION GENERALE	3
SECTION 2 : CONCEPTION DÉTAILLÉE	8
CONCLUSION PARTIELLE.....	12
CHAPITRE 2 : OUTILS ET IMPLEMENTATION	13
INTRODUCTION PARTIELLE	13
SECTION 1 : OUTILS UTILISES	13
SECTION 2 : PRESENTATION DES INTERFACES	15
CHAPITRE 4 : TESTS	22
1. Les tests au niveau du frontend.....	22
2. Les tests du backend	23
CONCLUSION GÉNÉRALE	26

CHAPITRE 1 : ANALYSE ET CONCEPTION DE L'APPLICATION

Dans l'ingénierie logicielle, toute conception d'un projet informatique nécessite une analyse basée sur un bon fondement. De ce fait cela passe par la réalisation du cycle de vie du projet. Raison pour laquelle la conception représente une phase primordiale et déterminante dans la production d'un logiciel de haute qualité. Cependant, cette phase nous permettra de visualiser globalement le logiciel d'une manière logique, ensuite de décrire l'architecture générale que nous allons utiliser dans la partie implémentation de notre projet. Enfin, de détailler notre choix conceptuel à travers différents types de diagrammes.

SECTION 1 : CONCEPTION GENERALE

I. CYCLE DE VIE

1. Définition

Le cycle de vie d'un logiciel est l'ensemble de toutes les étapes allant de la naissance jusqu'à sa disparition. Cette façon de faire c'est avérer utile dans la conception, car elle offre la possibilité de valider le développement du logiciel, et la vérification de son processus de développement.

2. Etapes du cycle vie

Le cycle de vie propre à un projet de réalisation d'une application d'authentification des diplômes, comprend généralement les activités ci-dessous :

- **Analyse du besoin et faisabilité** : elle permet de recueillir et formaliser le maximum d'information et l'ensemble des contraintes puis l'estimation de la faisabilité des besoins du client.
- **Spécification du besoin** : elle consiste ici à analyser et à bien cerné les finalités du projet de la solution à mettre en place.
- **Conception générale** : il s'agit ici de représenter l'architecture générale du logiciel.

- **Conception détaillé** : elle consiste à définir et formaliser respectivement chaque partie du logiciel.
- **Implémentation ou codage** : elle consiste à traduire dans un langage de programmation toutes les fonctionnalités du logiciel préalablement établi dans la phase de conception.
- **Test unitaire** : les tests déterminent les bugs techniques, les bugs fonctionnels et la qualité du logiciel ils permettent de s'assurer que chaque fonctionnalité du logiciel fonctionne exactement tel que décrit dans la conception.
- **Test d'intégration** : elle consiste à se rassurer que le logiciel corresponde exactement au cahier de charge du projet afin de mettre à la disposition des utilisateurs le manuel d'utilisation.
- **Mise en production** : c'est le déploiement général du logiciel.
- **Maintenance** : elle correspond à toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

3. Modèle de cycle de vie d'un logiciel

a. Modèle de cycle de vie en cascade

Figure 1: MODÈLE DE CYCLE DE VIE EN CASCADE



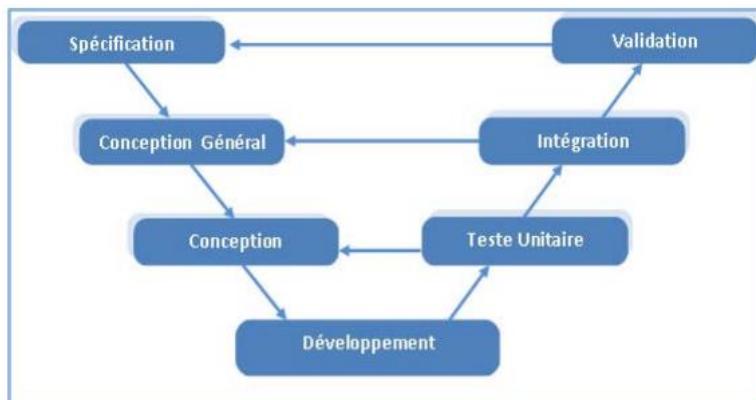
SOURCE : support de cour

Le model en cascade, ou « waterfall » en anglais, est une organisation des activités d'un projet sous forme de phases linéaires et séquentielles, où chaque phase correspond à une spécialisation des tâches et dépend des résultats de la phase précédente. C'est à dire que chaque phase se termine à une date précise par la production de certains documents ou logiciels.

L'inconvénient majeur du modèle de cycle de vie en cascade est que, la vérification du bon fonctionnement du système est réalisée trop tardivement : lors de la phase d'intégration, ou pire, lors de la mise en production.

b. Modèle de cycle de vie en V

Figure 2: MODÈLE DE CYCLE DE VIE EN V



SOURCE : Support de cours

Le cycle de vie en V est un modèle d'organisation des activités d'un projet qui se caractérise par un flux d'activité descendant qui détaille le produit jusqu'à sa réalisation, et un flux ascendant, qui assemble le produit en vérifiant sa qualité. Il demeure actuellement le cycle de vie.

Les avantages du modèle de cycle de vie en V sont les suivants :

- La qualité de la mise en œuvre des tests.
- Modèle éprouvé dans l'industrie.
- Normalisé (ISO-12207, MILSTD-498...).
- Deux types de tâches sont réalisées en parallèle : Verticalement on prépare l'étape suivante et Horizontalement : on prépare la vérification de la tâche en cours.

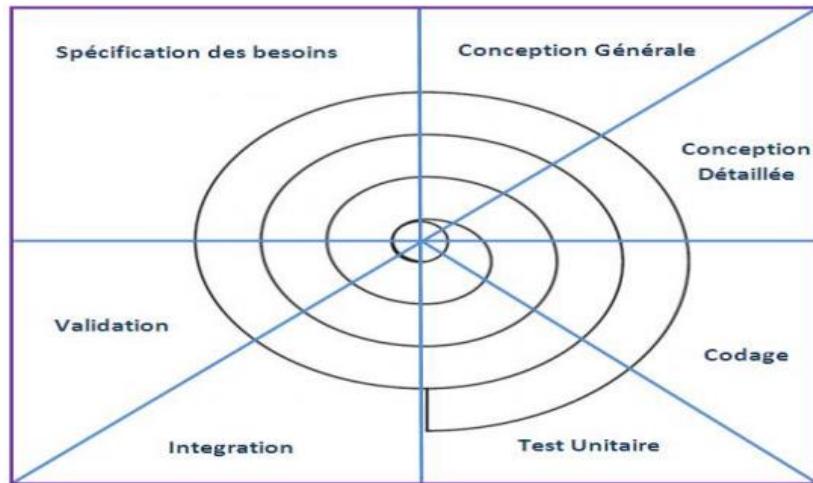
Ces inconvénients sont :

- La validation finale par le client très tardive augmente les risques de dépassement de délai et donc l'augmentation du coût.
- Phases séquentielles.

- Rigidité face à une évolution du besoin.

c. Modèle de cycle de vie en spirale

Figure 3: MODÈLE DE CYCLE DE VIE EN SPIRALE



SOURCE : Support de cour

Le modèle en spirale est un modèle de cycle de vie qui reprend les différentes étapes du cycle de vie en V. Par implémentation des versions successives, puis le cycle recommence en proposant un produit de plus en plus complet. Il met cependant plus l'accent sur la gestion des risques que le cycle en V.

4. Notre choix

Afin de concevoir et développer notre logiciel, nous avons opté pour le modèle de cycle de vie en Cascade. Pour des raisons tel que : simplicité et clarté, bonne documentation car elle est produite à chaque étape, adaptés aux petits projets et un suivi du planning facile.

II. MÉTHODOLOGIES DE CONCEPTIONS

Pour parfaire les fonctionnalités que doivent offrir le logiciel, La modélisation de ce projet s'est vue implanté par la méthode UML.

UML : est une méthode de conception et de réalisation des systèmes d'information en représentant les interactions entre différents composants à travers les diagrammes existant dans

le langage. La méthode UML préconise trois diagrammes à savoir : **Le diagramme des classes**, **Le diagramme des cas d'utilisation**, **Le diagramme des séquences**.

III. CONCEPTS ET ARCHITECTURE

Notre travail consiste à concevoir et à réaliser un logiciel d'authentification des diplômes. Etant donné que notre projet a été fait en groupe il a fallu que l'on s'oriente vers un modèle nous permettant de séparer ouvertement le back end du front end.

En se basant sur le model MVC (modèle vue contrôleur) qui est en une architecture interne à une application. Elle structure le code serveur pour séparer : le modèle c'est-à-dire les données et logique métier, Vue c'est-à-dire affichage, Contrôleur c'est dire le lien entre vue et modèle. Tandis que en adoptant l'API REST est orienté communication entre systèmes via HTTP. Elle ne gère pas l'affichage elle fournit uniquement des données, elle permet de découpler totalement le frontend du backend.

Notre choix se voie donc porter sur le modèle APIREST car il permet :

- découplage complet** : front end et back end peuvent être développés séparément.
- multi-client** : une API REST peut être utilisée par un site web, une application mobile, etc.
- Scalabilité** : plus facile à faire évoluer et à distribuer.

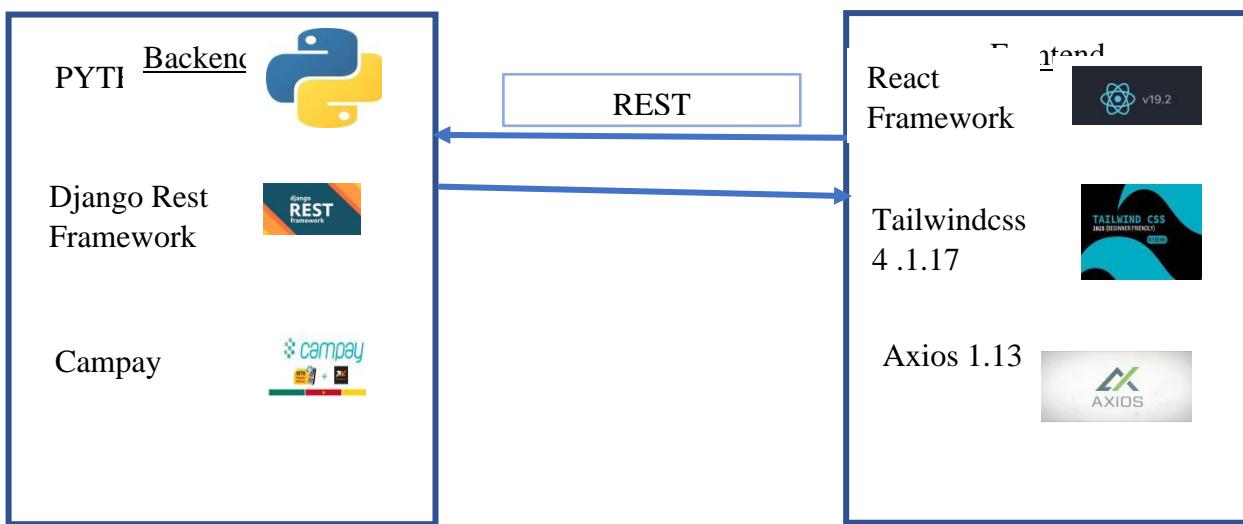


Figure 4: ARCHITECTURE DU MODÈLE APIREST

SECTION 2 : CONCEPTION DÉTAILLÉE

I. DIAGRAMME DES CAS D'UTILISATIONS

1. Identification des acteurs

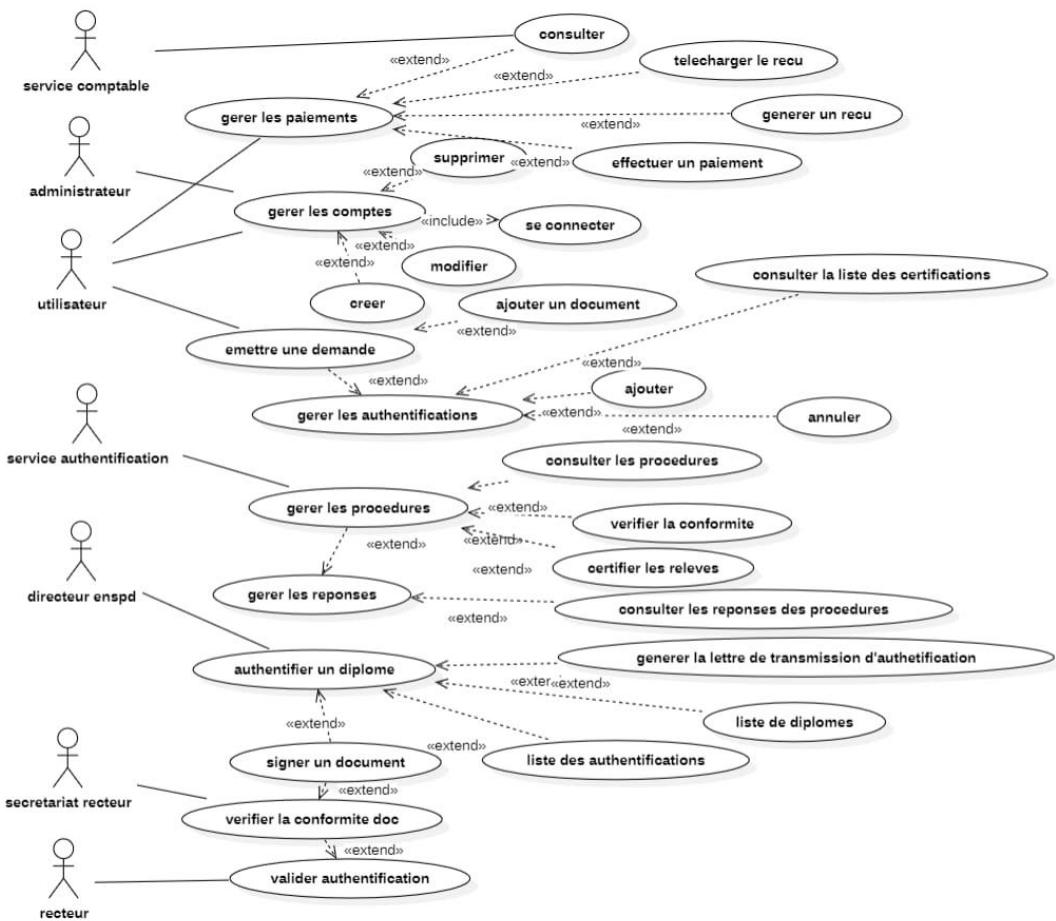
Tableau 1: ACTEUR INTERVENANT DANS LE SYSTÈME

Acteur	Tâches
Agent comptable	<ul style="list-style-type: none">★ Gérer les reçus.★ Gérer les comptes
Utilisateur	<ul style="list-style-type: none">★ Effectuer un paiement, émettre une demande★ Consulter les relevés certifiés.
Directeur	<ul style="list-style-type: none">★ Consulter les demandes d'authentification et de certification.★ Consulter la liste des diplômes authentifiés.★ Gérer les authentications de documents.

2. Elaboration du diagramme des cas d'utilisation

Les acteurs du système étant énumérés, Nous vous présenterons le diagramme de cas d'utilisation qui présente une vue globale du comportement fonctionnel de notre application. La figure ci-dessous présente le diagramme de cas d'utilisation globale de l'application.

Figure 5:DIAGRAMME DES CAS D'UTILISATION GOBAL



SOURCE : par nos soins

3. Description textuelle des cas d'utilisation

Cette description doit nous permettre de décrire exactement la façon dont un acteur utilise l'application pour atteindre l'objectif.

- Description du cas d'utilisation : **Gérer les comptes.**

Nom du UC	Gérer les comptes
Objectifs	Créer un compte, modifier les informations du compte, supprimer un compte
Auteurs concernés	Utilisateur
Précondition	Lancer le CDAS
Scenario nominal	<ul style="list-style-type: none">❖ Créer un compte❖ Cliquer sur l'onglet d'inscription❖ Remplir les informations demandées

	<ul style="list-style-type: none"> ❖ Valider l'inscription
Scenario alternatif 1	<ul style="list-style-type: none"> ❖ Cliquer sur l'onglet compte ❖ Choisir l'option modifier les informations ❖ Remplir les nouvelles informations que l'on souhaite mettre à jour ❖ Cliquer sur le bouton enregistrer
Scenario alternatif 2	<ul style="list-style-type: none"> ❖ Cliquer sur l'onglet compte ❖ Choisir l'option supprimer mon compte ❖ Valider
Post condition	

➤ Description du cas d'utilisation : **Gérer les authentifications**

Nom du UC	Gérer les authentifications
Objectifs	Ajouter une demande d'authentification, annuler une demande d'authentification, consulter les demandes déjà effectuées
Auteurs concernés	Utilisateur
Précondition	Lancer le CDAS et se connecter
Scenario nominal	<ul style="list-style-type: none"> ❖ Ajouter une demande d'authentification ❖
Scenario alternatif 1	<ul style="list-style-type: none"> ❖ Annuler une authentification ❖
Scenario alternatif 2	<ul style="list-style-type: none"> ❖ Consulter les demandes
Post condition	

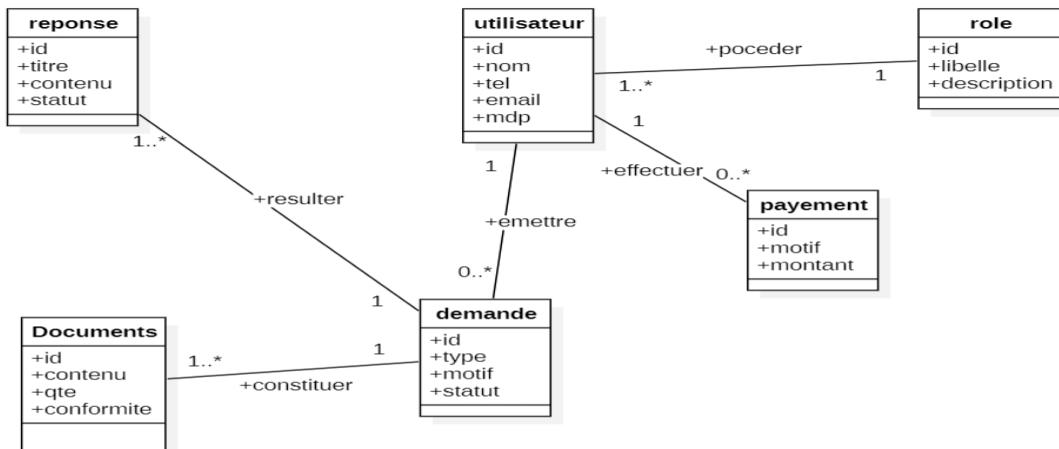
II. DIAGRAMME DES CLASSES

Le diagramme de classe est une représentation statique des objets et des éléments d'un système ainsi que les différentes relations entre celles-ci. Le diagramme suivant représente les

éléments et la structure de la base de données de notre logiciel. C'est un diagramme de classe abstrait il représente en générale les entités de notre application :

- ◆ **User** : classe qui représente les utilisateurs de l'application.
- ◆ **Demande** : c'est la classe qui prend en compte l'ensemble des demandes émises par les utilisateurs.
- ◆ **Réponse** : c'est la classe qui prend en compte l'ensemble des réponses envoyées aux utilisateurs par le système.
- ◆ **Paiement** : c'est la classe qui permet de garder une trace des différents paiements effectués.
- ◆ **Reçus** : c'est la classe qui permet de garder une trace des différents reçus émis lors des processus.
- ◆ **Comptabilité** : c'est la classe qui prend en compte les ventes effectuées dans l'entreprise.
- ◆ **Directeur** : c'est la classe qui stocke toutes les factures générées dans l'application.

Figure 6: DIAGRAMME DES CLASSES GLOBALES



SOURCE : par nos soins

- ❖ L'agence peut émettre une demande d'authentification, il peut également annuler la demande, consulter les demandes déjà émises ;
- ❖ Tous les utilisateurs peuvent gérer leur compte : créer un compte(inscription), modifier les informations du compte, ou supprimer son compte.
- ❖ Le service comptable peut gérer les paiements : consulte les paiements déjà effectués, générer un reçu de paiement

- ❖ L'agence peut effectuer un paiement et télécharger le reçu
- ❖ Le service d'authentification peut gérer les procédures : consulter la liste des procédures, gérer la conformité de la demande d'authentification, certifier les relevés de notes
- ❖ Le service d'authentification peut gérer les réponses en rapports avec les procédures, il peut ajouter une réponse, consulter les réponses
- ❖ Le directeur de l'ENSPD peut gérer les authentifications des diplômes : authentifier un diplôme, générer les lettres de transmission d'authentification, consulter les authentifications.
- ❖ Un utilisateur peut avoir 1 seul rôle (acteur externe=agence, service comptable, service authentification, directeur enspd)
- ❖ Un utilisateur (agence) peut lancer plusieurs procédures, une procédure n'est lancée que par une seule agence
- ❖ Une procédure peut posséder plusieurs documents, un document ne peut constituer qu'une seule procédure
- ❖ Une réponse ne peut résulter qu'une seule procédure (en fonction de sa conformité et de son résultat) elle est adaptée à chaque procédure, d'une procédure peut résulter plusieurs réponses (en fonction de la conformité et du type de procédure)

Un utilisateur (agence) peut effectuer plusieurs paiements, un paiement n'est effectué que par un seul utilisateur.

CONCLUSION PARTIELLE

Nous venons de terminer cette partie de conception, qui consiste à déterminer aussi bien les méthodes de travail à mettre en place pour formaliser les différentes fonctionnalités propres à l'application. Dans le chapitre suivant nous allons aborder la partie réalisation proprement dite de notre application, en se basant sur les mécanismes et les solutions déterminés dans la phase de conception.

CHAPITRE 2 : OUTILS ET IMPLEMENTATION

INTRODUCTION PARTIELLE

Ce chapitre présente la solution informatique réalisé. En effet, Nous présenterons dans un premier temps les outils utilisés au cours du développement et dans un deuxième temps la solution informatique proprement dites.

SECTION 1 : OUTILS UTILISÉS

1. ENVIRONNEMENT MATERIEL

L'environnement de travail est l'ensemble des outils, logiciel et langage utilisés pour l'implémentation d'une solution informatique.

Pour développer cette application nous avons utilisé des lap tops, configurées comme suit :

- Ordinateur dell latitude E6430.
- Mémoire vive : 6 Go
- Disque dur : 320 Go
- Processeur : Intel Core i5-3380M CPU 2.90GHZ ;
- Windows 10 professionnel

2. ENVIRONNEMENT LOGICIEL

1. LANGAGE ET TECHNOLOGIE UTILISÉ

- a. React Framework
 - **Type :** Front end
- **Rôle :** Développer des **Interfaces utilisateurs dynamiques (UI)**, principalement pour des applications web SPA (Single Page Applications).
- b. Tailwind ccs 4.1.17
- **Type :** Front end

- **Rôle** : Framework css utilitaire pour créer rapidement des interfaces **modernes et responsives** sans écrire trop de css personnalisé.
- **Avantages** : légers, rapide, très personnalisable, bien intégré avec les composants.

c. Axios 1.13

- **Type** : Front end
- **Rôle** : Framework css utilitaire pour créer rapidement des interfaces **modernes et responsives** sans écrire trop de css personnalisé.
- **Avantages** : légers, rapide, très personnalisable, bien intégré avec les composants

d. Python

- **Type** : Back end
- **Rôle** : Langage principal utilisé pour développer la **logique serveur**, l'API, les scripts.
- **Avantages** : Facile à apprendre, puissant, très utilisé en développement web, IA, data, etc.

e. Django REST Framework 3.14.0

- **Type** : Back end
- **Rôle** : Permet de créer des API RESTful puissantes en Python.
- **Avantage** : Authentification, permissions, sérialisation, pagination, gestion des requêtes http.

f. Campay

- Type : API externe(paiement)
- Rôle : Service d'intégration de **paiement mobile** au Cameroun
- Utilisation : Back end consomme l'api campay pour initier, suivre ou confirmer des paiements.

2. IDE ET ETUDE DE CODE : VISUAL STUDIO

Un éditeur de code source est un programme d'édition de texte spécialement conçu pour l'édition de code d'un programme informatique. Il en existe plusieurs à savoir :

- Visual studio
- Eclipse
- Visual basic
- NetBeans

Cependant nous avons opté pour Visual studio qui est un éditeur de code extensible et très flexible car il inclut :

- Un système de débogage.
- La mise en évidence de la syntaxe.
- La complétion intelligente.
- Le nettoyage du cache des projets.
- La libération de l'espace mémoire.
- Etc...

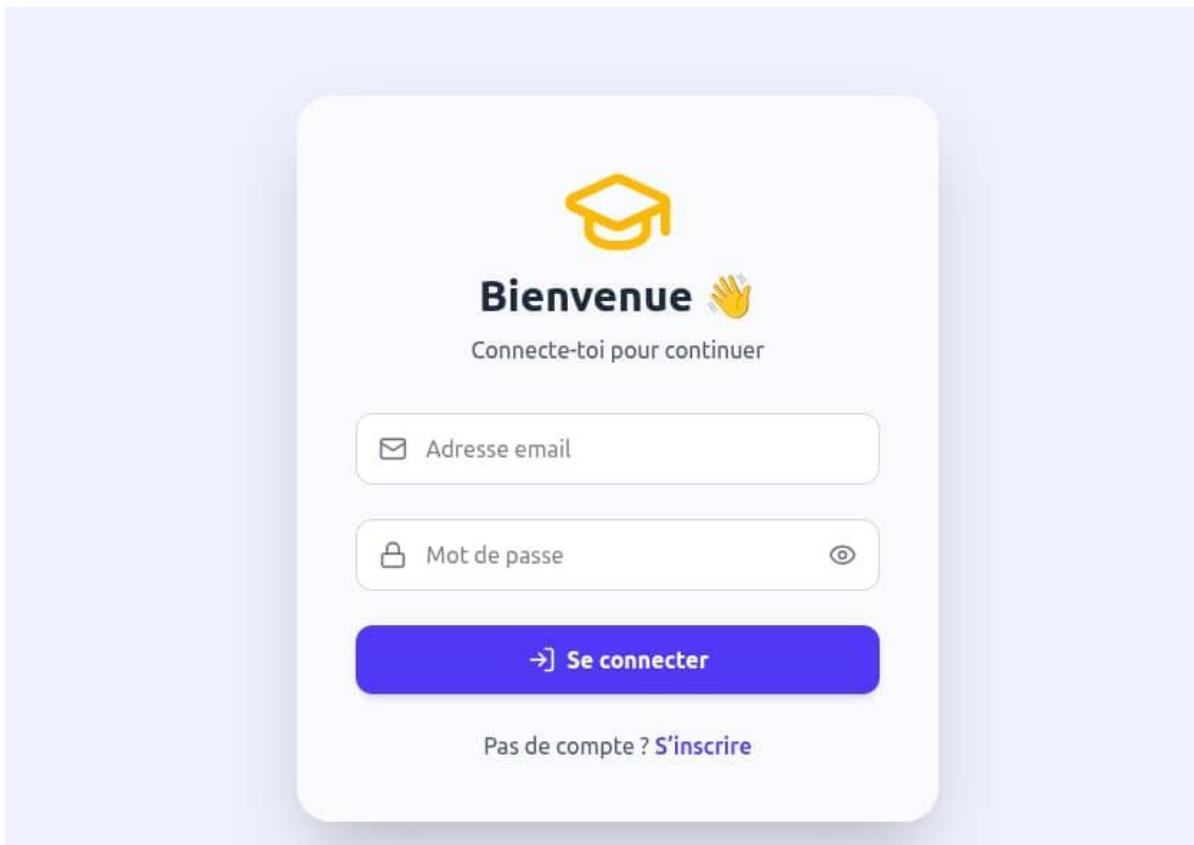
SECTION 2 : PRESENTATION DES INTERFACES

Dans cette partie, nous allons vous présenter les interfaces principales de notre projet. Application Web, elle permettra d'enregistrer la différente information relative soit à un client, un produit, ou un utilisateur.

a. Authentification

La figure suivante présente l'interface d'authentification où doit passer tous les utilisateurs afin de se connecter à l'application :

Figure 7: INTERFACE DE CONNEXION DE L'APPLICATION



Source : *par nos soins*

L'administrateur et le super administrateur avant d'effectuer une opération dans le logiciel doivent au préalable s'authentifier à travers le formulaire ci-dessus au cas échéant il n'aura pas accès au logiciel.

b. Page principale

Les figures suivantes présentent l'interface se présentant lors de l'ouverture de l'application. Elle décrit en résumé ce que fait l'application.

Figure 8: INTERFACE PRINCIPALE



Source : par nos soins

c. Interface d'inscription

La figure suivante vous présente l'interface permettant d'inscription d'un nouvel utilisateur ainsi que les différentes opérations susceptibles d'être appliquée sur celui-ci.

Figure 9: INTERFACE DE GESTION DES UTILISATEURS

The screenshot displays the 'Créer un compte' (Create account) form. At the top, there is a button for 'Continuer avec Google'. Below it, a line of text says 'ou créer un compte avec email'. The form contains five input fields: 'Nom complet' (with a person icon), 'Numero de téléphone' (with a phone icon), 'Adresse email' (with an envelope icon), 'Mot de passe' (with a lock icon), and 'Confirmer le mot de passe' (with a lock icon). To the right of each password field is an 'eye' icon for password visibility. At the bottom of the form is a large blue 'S'Incrire' (Sign up) button. Below the button, a link reads 'Déjà un compte ? Se connecter'.

Source : par nos soins

d. Interface de l'application

Les figures ci-dessous vous présente les interfaces de gestion globale du logiciel. Le processus d'enchaînement est le suivant : on débute sur le formulaire de déclaration de la procédure ici c'est la première interface où il choisit le type de procédure qu'il souhaite démarrer.

Figure 10: INTERFACE DU CHOIX DE PROCEDURE

Nouvelle Procédure

TYPE DE REQUETES INFORMATIONS PERSONNELLES TÉLEVERSEMENT DE VOS DOCUMENTS VÉRIFICATION DU PAYEMENT VALIDATION DU PAYEMENT DERNIÈRE VÉRIFICATION ET CONFIRMATION

Type de procédure

-- Sélectionner une procédure --

Motif de la demande

Ex : procédure de voyage, demande d'emploi, poursuite d'études...

REVENIR EN ARRIÈRE SUIVANT

Source : par nos soins

Puis précise les informations sur l'étudiant.

Figure 11: INTERFACE DE RENSEIGNEMENT

The screenshot shows a step-by-step process titled "Nouvelle Procédure". The steps are represented by circles numbered 1 through 6. Steps 1 and 2 are completed (blue circles with checkmarks), while steps 3 through 6 are in progress (yellow circles). Step 3 is labeled "INFORMATIONS PERSONNELLES".

Below the steps, there are four input fields arranged in a 2x2 grid:

- Matricule:** Ex : 24G093454
- Nom:** Ex : Mamba
- Prénom:** Ex : Ozone
- Filière:** Ex: Genie informatique
- Ecole:** Ex: ENSPD
- Moyenne Générale:** 00

At the bottom left is a "REVENIR EN ARRIÈRE" (Back) button, and at the bottom right is a blue "SUIVANT" (Next) button.

Source : par nos soins

Ensuite il charge les documents qui devront être traité.

Figure 12: INTERFACE DE CHARGEMENT DES DOCUMENTS

This screenshot shows the same "Nouvelle Procédure" interface, but the current step is "TÉLEVERSEMENT DE VOS DOCUMENTS" (Step 4). The progress bar indicates steps 1, 2, and 3 are completed, while 4, 5, and 6 are in progress.

A large dashed rectangular area in the center contains an upward arrow icon and the text "Cliquez pour ajouter des documents PDF, JPG, PNG (plusieurs fichiers autorisés)".

At the bottom left is a "REVENIR EN ARRIÈRE" button, and at the bottom right is a blue "SUIVANT" button.

Source : par nos soins

Cette interface présente le montant qu'il aura à payer.

Figure 13: INTERFACE POUR LE MONTANT A PAYER

Nouvelle Procédure

TYPE DE REQUETES INFORMATIONS PERSONNELLES TÉLEVERSEMENT DE VOS DOCUMENTS VÉRIFICATION DU PAIEMENT DERNIÈRE VÉRIFICATION ET CONFIRMATION

Type de procédure
Prix unitaire **0 FCFA**

Total à payer **0 FCFA**

Numéro de paiement
Entrez votre numéro Mobile Money
Le paiement sera initié via ce numéro. Assurez-vous qu'il soit correct.

REVENIR EN ARRIÈRE **SUIVANT**

Source : par nos soins

En fin, il vérifie si les informations qu'il a entrées est correcte avant de soumettre son formulaire.

Figure 14: FORMULAIRE A SOUMETTRE

Nouvelle Procédure

TYPE DE REQUETES INFORMATIONS PERSONNELLES TÉLEVERSEMENT DE VOS DOCUMENTS VÉRIFICATION DU PAIEMENT DERNIÈRE VÉRIFICATION ET CONFIRMATION

Récapitulatif de la demande

Procédure
Type : authentification
Motif : hfoiiuwf

Informations personnelles
hiohoigh hsoih
Email :
Téléphone :

Documents
1 document(s) téléchargé(s)

Paiement
Total payé : **25 000 FCFA**

Toutes les informations sont prêtes à être soumises

REVENIR EN ARRIÈRE **VALIDER**

e. Interface pour le tableau de bord

L’interface suivant est le tableau de bord, montrant l’état global des différents processus. C’est-à-dire permet de connaitre les processus en cours et ceux terminé.

FIGURE 28 : TABLEAU DE BORD

The screenshot shows a dashboard titled "CDAS Diplomas". At the top, there is a button labeled "+ Nouvelle procédure". Below it, a table titled "Liste de vos procédures" displays two rows of data:

N°	TYPE DE PROCÉDURE	CONCERNÉ(E)	MOTIF	STATUT
1	Authentification	Mamba ozone	proc voyage	En cours
2	Certification	Mamoud van	proc voyage	Annulé

Source : par nos soins

Dans ce chapitre de réalisation, nous avons été appelés à présenter les outils et moyen utiliser pour aboutir à la concrétisation de la solution informatique étudier plus haut ; ainsi que de présenter les différentes interfaces de la dites application.

CHAPITRE 4 : TESTS

1. Les tests au niveau du frontend

Les tests au niveau de la partie visible de l'application ont été réalisés sur l'éditeur de code VS Code à l'aide de **Vitest + React Test Library**. Ces tests nous ont permis de mieux voir nos fonctionnalités et d'apporter des corrections au niveau du code source de ses tests résultent les figures suivantes qui résument les fonctionnalités qui ont été testé et les résultats finaux :

```
vitest src/components/intro.test.jsx      coverage

DEV v4.0.16 /home/mamoud/Bureau/CDAS/cdas_frontend

✓ src/components/intro.test.jsx (3 tests) 184ms
  ✓ Intro Component (3)
    ✓ rend correctement l'image, le titre et les boutons 125ms
    ✓ navigue vers la page login au clic sur Se Connecter 36ms
    ✓ navigue vers la page register au clic sur S'inscrire 19ms

Test Files 1 passed (1)
Tests 3 passed (3)
Start at 10:15:47
Duration 2.06s (transform 168ms, setup 142ms, import 501ms, tests 184ms,
environment 959ms)

PASS Waiting for file changes...
press h to show help, press q to quit
```

```

DEV v4.0.16 /home/mamoud/Bureau/CDAS/cdas_frontend
Coverage enabled with v8

✓ src/components/form_connex.test.jsx (5 tests) 298ms
✓ Login Component (5)
  ✓ affiche correctement les champs email et mot de passe 104ms
  ✓ permet de saisir email et mot de passe 42ms
  ✓ appelle login lors de la soumission du formulaire 60ms
  ✓ permet de basculer l'affichage du mot de passe 38ms
  ✓ redirige après un login réussi 49ms

Test Files 1 passed (1)
  Tests 5 passed (5)
  Start at 03:31:21
  Duration 2.49s (transform 233ms, setup 153ms, import 622ms, tests 298ms,
environment 945ms)

% Coverage report from v8
-----|-----|-----|-----|-----|-----|
File    | % Stmt | % Branch | % Funcs | % Lines | Uncovered Line #
-----|-----|-----|-----|-----|-----|
All files | 85.71 | 64.28 | 100 | 85.71 |
..._connex.jsx | 85.71 | 64.28 | 100 | 85.71 | 32-35
-----|-----|-----|-----|-----|-----|
PASS Waiting for file changes...
press h to show help, press n to quit

```

2. Les tests du backend

Comme pour les interfaces, nous avons mené des tests au niveau de notre api que nous avons mis en place ; les images suivantes nous présentent ce qui en résulte :

The screenshot shows the PyCharm IDE interface with the following details:

- Project Structure:** The project is named "CDAS" and contains a "backend" directory. Inside "backend", there is a "accounts" directory which includes "migrations", "admin.py", "apps.py", "models.py", "serializers.py", "services.py", "tests.py", and "views.py".
- Code Editor:** The "tests.py" file is open in the main editor window. It contains test cases for "UtilisateurRegisterSerializer" and "UtilisateurSerializer".
- Terminal:** The terminal shows a system check command running, identifying one issue related to ACCOUNT_LOGIN_METHODS.
- Status Bar:** The status bar at the bottom indicates "Ran 12 tests in 0.378s".

The screenshot shows the PyCharm IDE interface with the following details:

- Project Structure:** The project is named "CDAS" and contains a "backend" directory. Inside "backend", there is a "procedures" directory which includes "migrations", "admin.py", "apps.py", "campus_clients.py", "models.py", "serializers.py", "services.py", "signals.py", "tests.py", "traitement_document.py", "utils.py", and "views.py".
- Code Editor:** The "tests.py" file is open in the main editor window. It contains a test case for "ProcedureTests".
- Terminal:** The terminal shows a system check command running, identifying one issue related to ACCOUNT_LOGIN_METHODS.
- Status Bar:** The status bar at the bottom indicates "Ran 5 tests in 3.513s".

The screenshot shows the PyCharm IDE interface with two code editors open. The left editor displays `tests.py` which contains the following code:

```
from rest_framework import serializers
from .models import Utilisateur, Role
from .services import UtilisateurService

class UtilisateurRegisterSerializer(serializers.ModelSerializer):
    nom = serializers.CharField(max_length=100)
    email = serializers.EmailField()
    telephone = serializers.CharField(max_length=20)
    password = serializers.CharField(write_only=True)
    role_id = serializers.IntegerField()

    def create(self, validated_data):
        return UtilisateurService.create_utilisateur(validated_data)

class UtilisateurSerializer(serializers.ModelSerializer):
    role = serializers.StringRelatedField()
    role_id = serializers.PrimaryKeyRelatedField(
        queryset=Role.objects.all(), source="role", write_only=True
    )

    class Meta:
        pass
```

The right editor displays `accounts/tests.py` which contains test cases for permissions:

```
class UtilisateurTests(APITestCase):
    def test_role_permission_check(self):
        perm = Permission.objects.create(
            codename="can_test",
            description="Test permission"
        )
        self.role.permissions.add(perm)
        self.assertTrue(self.admin.has_permission("can_test"))
        self.assertFalse(self.admin.has_permission("unknown_permission"))
```

CONCLUSION GÉNÉRALE

Durant notre cheminement, nous avons développé un logiciel d'authentification des diplômes pour l'école national Supérieur Polytechnique de Douala.

Ce logiciel répond aux besoins suivants : Authentifier les diplômes, Certifier les diplômes, Répondre aux demandes. En sommes la mise en œuvre de cette solution permettra a l'école polytechnique de Douala de gagner en temps.