

Rapport de projet TPI

Gestionnaire de notes de frais

Réalisé par:

AGREBI Jihene

AHMED Saad el din

BEN EL BEY Yessine

NORTIER Hugo

Année universitaire: 2022-2023

Table des matières

Table des matières	1
Résumé	2
Introduction	4
I. Etude préalable	6
1. Cadre général du projet	6
2. Présentation de l'entreprise	6
3. Présentation de l'équipe	6
4. Présentation du sujet et objectifs du projet	6
5. Etat de l'art	7
5.1. Analyse des librairies OCR existantes	7
5.2. Recherche sur les technologies concernées	8
Computer vision	8
Machine/Deep learning	9
5.3. Les autres applications de note de frais	9
6. Etude de l'existant et solution envisagée	12
II. Démarche projet	13
1. Méthode de gestion de projet utilisé	13
2. Rôles et responsabilités	14
3. Outils Utilisés	15
4. Coût du projet	16
5. Risques du projet	17
III. Exigences fonctionnelles	18
IV. Architecture système	19
Architecture logicielle	19
V. Réalisations	20
1. Application mobile	20
Description	20
Conception du logiciel développé	20
Réalisation	21
2. API OCR	24
Traitement initial de la requête	25
3. Site web	26
Backend	26
Frontend	30
Conclusion	41
Références	43

Résumé

Pour aider les gestionnaires de l'université dans leur tâches comptables au quotidien, nous avons été chargés de mettre en place une architecture numérique qui permettra de leur faciliter le travail pour le traitement des factures et la génération de notes de frais de chaque collaborateur.

Après avoir étudié l'existant des groupes qui ont déjà travaillé sur le sujet les années précédentes, nous avons développé plusieurs applications pour répondre à la problématique:

- un module ocr en python qu'à partir d'une photo de facture va essayer d'extraire les informations clé (le siret, totaux tva, ttc, ht, date de la facture et la catégorie)
- une application mobile android développée en Kotlin qui permettra aux collaborateurs de scanner leurs factures et, une fois validé les données à sauvegarder, les transmettre au service de gestion.
- un site web en ligne développé en React pour le front et en Node pour le back qui permettra aux gestionnaires de gérer les profils des collaborateurs et leurs factures ainsi que pouvoir générer des notes de frais sur des périodes précises.

Ce projet nous a offert une opportunité précieuse d'élargir nos connaissances dans divers domaines, notamment le machine learning et la vision par ordinateur. Nous avons également pu améliorer notre compétence en matière de développement d'architectures complexes avec plusieurs applications intégrées.

Abstract

To assist the university managers in their daily accounting tasks, we were tasked with implementing a digital architecture that will simplify their work for processing invoices and generating expense reports for each collaborator.

After studying the existing solutions from groups that have worked on the subject in previous years, we developed several applications to address the issue:

- A Python OCR module that, from a photo of a bill, will try to extract key information ("SIRET", "total TVA", "TTC", "HT", "factures", "date", and "categorie").
- An Android mobile application developed in Kotlin that will allow collaborators to scan their invoices and, once validated and saved, transmit the data to the management service.
- An online web platform developed in React for the front-end and Node for the back-end, which will enable managers to manage collaborator profiles and their invoices, as well as generate expense reports for specific periods.

This project offered us a valuable opportunity to expand our knowledge in various fields, including machine learning and computer vision. We were also able to improve our skills in developing complex architectures with multiple integrated applications.

Introduction

La gestion des notes de frais est une tâche ardue et routinière qui monopolise une grande partie du temps des employés dans de nombreuses entreprises. Cette tâche implique la saisie de factures, la collecte de justificatifs et la création de fichiers pour les services comptables. Malheureusement, cette tâche est souvent réalisée manuellement, ce qui peut être chronophage et peut également conduire à des erreurs de saisie ou à des pertes de documents.

Malgré l'existence de processus et d'applications pour faciliter la gestion des notes de frais dans certaines entreprises, cela n'est malheureusement pas le cas pour l'UCA. Les assistantes administratives de l'UCA consacrent donc une quantité considérable de temps à effectuer cette tâche manuellement, ce qui peut nuire à leur productivité et à leur capacité à se concentrer sur d'autres tâches importantes. Actuellement, l'unique outil existant pour faciliter la tâche est un modèle de fichier Excel qui doit être rempli manuellement avec les informations récupérées.

Cependant, il existe des solutions pour simplifier la gestion des notes de frais. Par exemple, les entreprises peuvent utiliser des logiciels de gestion de notes de frais pour automatiser le processus de saisie des factures et de collecte de justificatifs. Cela permet non seulement de gagner du temps, mais également de réduire les risques d'erreurs et de pertes de documents. C'est pourquoi l'UCA envisage de mettre en place une application web, une application mobile et un OCR pour automatiser le processus de récupération des justificatifs et de construction des documents justificatifs.

En résumé, la gestion des notes de frais est une tâche administrative cruciale qui peut prendre beaucoup de temps si elle est effectuée manuellement. Cependant, en utilisant des outils et des processus efficaces, les entreprises peuvent simplifier cette tâche et améliorer la productivité de leur personnel. L'UCA souhaite ainsi mettre en place une solution d'automatisation pour faciliter la gestion des notes de frais et améliorer l'efficacité de son personnel.

Le suivant rapport est subdivisé en 5 parties, qui décrivent le travail effectué, comme suit:

- La première section consistera à introduire l'étude préliminaire qui comprendra une présentation de l'entreprise et du sujet du TPI, ainsi qu'une présentation de l'équipe. Nous allons également passer en revue les travaux existants et effectuer une analyse de l'état de l'art.
- La deuxième section couvrira la gestion de projet, y compris la méthode employée, l'étude des risques ainsi que le budget du projet.

- La troisième section abordera les exigences fonctionnelles de notre projet.
- La quatrième section consistera à présenter l'architecture logicielle globale de notre projet.
- La dernière partie décrira nos réalisations en présentant une conception des codes sources et des interfaces réalisées.

I. Etude préalable

1. Cadre général du projet

Le projet de gestion des notes de frais proposé par l'UCA a été une opportunité pour nous d'appliquer les connaissances et les compétences acquises au cours de notre cursus universitaire et professionnel dans le cadre d'un projet de thèse d'innovation.

2. Présentation de l'entreprise

L'**université Côte d'Azur** est une université française créée en 2019 et prenant la suite de l'université de Nice créée en 1965, encore mentionnée dans le code de l'éducation en 2018. Cette université est celle dont notre cursus MIAGE fait partie.

MIAGE ou (Méthodes Informatique Appliquées à la Gestion des Entreprises) est un master universitaire français connu par ses compétences de niveau Bac+5, alliant une double compétence en informatique et en gestion, destiné à former des cadres d'entreprise experts en ingénierie et management des systèmes d'information. Connu pour sa qualité de formation, l'accès à ces formations est particulièrement sélectif, via des concours et entretiens, après deux années d'études (DUT, BTS, Classes préparatoires, L2). La durée de la formation est normalement de trois ans et se termine par l'obtention d'un master. Elle est beaucoup plus basée sur deux grands axes (les technologies de l'information TIC et les sciences de gestion).

3. Présentation de l'équipe

Notre équipe est constituée de quatre membres, à savoir Jihene Agrebi, Saad el din Ahmed, Yessine Ben El Bey et Hugo Nortier. Nous sommes tous des étudiants de M2 MIAGE inscrits dans le cursus MBDS et nous sommes également des alternants en développement. Nous possédons des compétences techniques équilibrées qui nous permettent d'aborder ce projet avec confiance et de le mener à bien.

4. Présentation du sujet et objectifs du projet

L'Université doit actuellement traiter les notes de frais (NDF) de manière manuelle, en collectant les factures de chaque collaborateur chaque mois. Nous avons pour objectif de les aider et de faciliter leur travail en mettant en place un outil de gestion de NDF. Cet outil comprendra un backoffice web utilisé par les gestionnaires (admin), leur permettant ainsi de gérer les utilisateurs, leurs factures et de générer les NDF. Les données seront stockées dans une base de données accessible via une API REST.

Nous prévoyons également la création d'une application mobile pour les utilisateurs qui leur permettra de numériser leurs factures. Ensuite, un OCR sera utilisé pour extraire les

informations clés telles que la date de la facture, le montant hors taxe, le montant toutes taxes comprises, la TVA totale (5,5%, 10%, 20%) et la catégorie (par exemple : restauration). Si nécessaire, les gestionnaires pourront modifier les informations extraites par l'OCR si elles sont inexactes. En outre, nous prévoyons de créer un mode administrateur pour l'application mobile qui permettra aux gestionnaires de numériser les factures pour n'importe quel utilisateur.

5. Etat de l'art

5.1. Analyse des librairies OCR existantes

Afin de pouvoir lire les factures scannées par les utilisateurs, nous devons utiliser un OCR. Dans cette optique, nous avons procédé à une analyse comparative des différentes bibliothèques disponibles, afin de choisir celle qui convient le mieux à nos besoins.

Ocrad.js

Avantages :

- Rapide pour un OCR

Inconvénients :

- Fonctionne qu'avec un alphabet latin
- Fonctionne mal avec de l'écriture manuscrite

Tesseract.js

Avantages :

- Puissant et fiable dans la reconnaissance
- Fonctionne bien pour l'écriture manuscrite et des lettres « anciennes » et non régulières
- Librairie utilisée dans les anciens projets

Inconvénients :

- Traitement qui prend du temps

KerasOCR.js

Avantages :

- Facile à utiliser
- Déploiement rapide et simple

- Peut utiliser plusieurs GPU (faire plusieurs tâches, répartir la charge des projets)

Inconvénients :

- Très probable qu'il y ait des erreurs de l'API de bas niveau car nous sommes très limités dans la modification. Les journaux d'erreurs sont difficiles à déboguer.
- Traitement qui prend du temps

EasyOCR.js

Avantages :

- Fonctionne bien même avec des images floues et très fournies
- Rapide
- Renvoie la position exacte sur l'image du texte détecté

Inconvénients :

- Si un document contient des langues autres que celles indiquées dans les arguments du LANG, les résultats peuvent également être médiocres
- Pour du texte manuscrit, le résultat est très faible
- Des numérisations de mauvaise qualité peuvent générer de mauvais résultats

Suite à nos analyses, nous avons décidé de choisir EasyOCR. EasyOCR est le seul du lot qui nous permet de connaître exactement la position du texte détecté sur l'image, ce qui nous donnera l'opportunité de renvoyer aux utilisateurs le résultat du OCR et leur permettre de les corriger, ce qui réduit considérablement l'obligation d'avoir un OCR très performant. De plus, les inconvénients que possède cette librairie ne nous impactent pas, mise à part que la librairie risque de donner des prédictions faussées si l'image fournie est de mauvaise qualité mais ce risque peut être ignoré étant donné que c'est un cas rare et que l'utilisateur aura le choix de les corriger.

5.2. Recherche sur les technologies concernées

Computer vision

La vision par ordinateur est une branche de l'intelligence artificielle qui permet aux ordinateurs d'extraire des informations significatives à partir d'images numériques, de vidéos et d'autres entrées visuelles, en utilisant des techniques telles que le changement de couleur, le contraste et la mise en relief des images. Pour entraîner un modèle de vision par

ordinateur, nous utilisons l'apprentissage profond avec des réseaux de neurones convolutifs (CNN).

Dans notre cas, nous utilisons des OCR qui réalisent de la vision par ordinateur en s'appuyant sur ces technologies. Il faut donc, avant de pouvoir utiliser l'OCR, entraîner un modèle à la détection de texte avec un réseau de neurones. Les solutions déjà disponibles sur internet arrivent souvent avec des modèles déjà entraînés mais ça n'empêche pas que nous avons la possibilité d'effectuer davantage d'entraînement pour avoir de meilleurs résultats voire entraîner notre propre modèle à partir de rien pour qu'il détecte uniquement les données sur les factures dont nous avons besoin.

Machine/Deep learning

- L'OCR sert seulement à numériser du texte de documents et à le rendre lisible par une machine mais elle ne comprend et n'interprète aucune donnée. Pour cela, l'OCR va être complété par le machine learning.
- Le machine learning va plus loin que l'OCR car il va permettre de comprendre le texte extrait d'un document. Les algorithmes permettent d'interpréter le contenu et le comprendre afin d'analyser une facture d'une manière similaire à l'être humain. Elle utilise ainsi le résultat du traitement OCR, puis applique son propre traitement pour comprendre des mots et même des phrases, et à traiter une facture comme le ferait un être humain inconsciemment.

5.3. Les autres applications de note de frais

Voici une liste non exhaustive d'applications de notes de frais les plus populaires aujourd'hui.

Expensify

Elle permet aux utilisateurs de numériser les reçus de dépenses et de générer des rapports de dépenses. Elle permet également de demander l'approbation d'une dépense potentielle ou le règlement de cette dernière.

Quels sont les avantages d'utiliser expensify ?

Tout comme NDF app, expensify s'appuie sur la reconnaissance optique de caractères (OCR) pour lire et extraire les informations pertinentes à partir de reçus, de tickets de caisse ou encore de factures. Grâce à l'algorithme, ces informations sont ensuite automatiquement intégrées dans un rapport de dépenses. Par ailleurs, Expensify est équipé d'un « Smart Scan », qui est un dispositif permettant de numériser les reçus par le biais de l'appareil

photographique d'un smartphone. C'est donc une application à la pointe de la technologie qu'un autoentrepreneur peut utiliser en toute circonstance, à condition, bien sûr, qu'il n'ait pas oublié son smartphone.

Quel est le prix d'expensify ?

Expensify propose plusieurs plans tarifaires pour répondre aux besoins des particuliers ou des professionnels. Les plans tarifaires sont pour les plus simplifiés totalement gratuits, tandis que ceux nécessitant une personnalisation de l'application sont adaptés selon les spécifications apportées.

Zoho expense

Zoho Expense est une application de note de frais qui permet aux utilisateurs de suivre les dépenses, de gérer les reçus et de générer des rapports de dépenses.

Quelle est sa particularité en comparaison aux autres applications de notes de frais ?

La particularité de Zoho Expense est sa simplicité d'utilisation et sa facilité d'intégration avec les autres outils de la suite Zoho. En d'autres termes, un particulier ou une entreprise se tournant vers cette solution aura tout intérêt d'utiliser également les autres applications Zoho afin de rendre sa souscription rentable. Parmi ces applications, il est possible de compter une application de gestion des relations avec la clientèle, appelée Zoho CRM, et un outil de gestion de projet permettant de suivre les projets et les tâches, gérer les calendriers et les échéances d'un livrable, ou encore collaborer avec les autres membres de l'équipe. C'est donc une solution plurifonctionnelle qui peut être adaptée à des structures conséquentes comme des multinationales.

Quel est son prix ?

Tout comme Expensify, Zoho expense propose des grilles tarifaires pouvant s'adapter aux exigences de chacun.

Voici un aperçu des tarifs de Zoho Expense :

- Le plan gratuit s'adresse à 3 utilisateurs maximum, avec des fonctionnalités de base telles que la numérisation des reçus, la gestion des notes de frais et la création de rapports de dépenses.

- Le plan intermédiaire, ou plan standard, peut être obtenu à partir de 5 € par utilisateur et par mois. Il propose des fonctionnalités avancées telles que l'automatisation des dépenses, la gestion des approbations, la synchronisation avec la comptabilité, etc.
- Enfin, le plan professionnel, qui est le plus adapté aux entreprises, coûte 8 € par utilisateur et par mois. Le plan professionnel offre des fonctionnalités avancées telles que les politiques de dépenses personnalisées, l'importation en masse des transactions, etc.

Concur

Concur est une application de note de frais qui permet aux entreprises de gérer les dépenses des employés en un seul endroit.

Pourquoi se tourner vers Concur ?

Concur possède plusieurs fonctionnalités qui ne sont pas intégrées dans la majorité des applications de note de frais. Parmi ces fonctionnalités, Concur offre la possibilité de se connecter à des fournisseurs de voyages tels que les compagnies aériennes, les hôtels, les agences de location de voitures, etc. pour récupérer automatiquement les informations de voyage et les ajouter aux notes de frais. De plus, Concur permet de réaliser un traitement de dépenses en temps réel, ce qui a pour objectif de faciliter la gestion des employés. Concur est donc une application plus adaptée pour les employés se voyant régulièrement en voyages professionnels.

Quel est le prix de Concur ?

Malheureusement, il ne sera pas possible d'identifier la politique tarifaire utilisée par Concur, ses prix ne sont pas disponibles sur son site web. En effet, ces derniers peuvent varier en fonction des besoins spécifiques de chaque entreprise.

Concur est généralement considéré comme une solution destinée aux grandes entreprises, avec des coûts associés plus élevés que d'autres logiciels de gestion de notes de frais. Les tarifs sont basés sur différents critères tels que le nombre d'utilisateurs, le volume de transactions, les fonctionnalités nécessaires, etc.

Jenji

Jenji est également une application de note de frais qui permet aux entreprises de gérer facilement leurs dépenses professionnelles.

En quoi Jenji se distingue des autres applications de notes de frais ?

Elle offre plusieurs fonctionnalités, telles que la numérisation des reçus, la gestion des notes de frais, la création de rapports de dépenses et la synchronisation avec les logiciels de comptabilité.

Quel est le prix de jenji ?

Jenji propose une grille tarifaire à ses utilisateurs selon leurs attentes et leurs besoins vis-à-vis de l'application. En effet, chaque entreprise ou particulier a des attentes bien définies concernant une application de notes de frais, et voici comment Jenji tend à y répondre :

- Dans un premier temps, pour les particuliers ou les entreprises ne nécessitant pas une grande base de données, Jenji offre la possibilité de se tourner vers une solution gratuite. Dans ce cas, Jenji permet à un utilisateur de numériser jusqu'à 10 reçus par mois et de générer des rapports de dépenses illimités.
- Si cela est estimé par l'utilisateur comme étant trop succinct, Jenji propose à ses clients le plan standard. Le plan standard de Jenji commence à partir de 5€ par mois et par utilisateur, avec des fonctionnalités telles que la numérisation de reçus illimitée, la gestion des notes de frais, la création de rapports de dépenses et l'intégration avec les logiciels de comptabilité. Les fonctionnalités proposées par l'application sont donc plus complètes et l'utilisation de la base de données n'est pas limitée.
- Concernant les entreprises, Jenji propose un plan entreprise personnalisé pour répondre à des besoins spécifiques en matière de gestion des dépenses. Les tarifs pour ce plan sont disponibles sur demande.

6. Etude de l'existant et solution envisagée

Deux projets ont été réalisés par des étudiants les années précédentes. Nous avons entrepris une analyse approfondie du code pour déterminer leur viabilité. Étant donné que le deuxième projet a repris le travail effectué sur le premier projet, nous avons axé notre étude sur ce dernier. Cependant, nous avons identifié plusieurs limites qui nous ont poussés à ne pas reprendre les applications existantes. En effet, elles ont été développées avec des technologies que nous ne maîtrisons pas (Flutter pour mobile et Angular pour le back office)

et la qualité du code n'est pas satisfaisante. Nous sommes donc décidés à tout refaire depuis le début avec nos propres moyens afin de gagner en rapidité.

Malgré cela, nous nous inspirerons de leurs approches pour l'analyse et l'extraction de données. Nous reprendrons notamment le fichier Bill.js qui effectue des manipulations pour retrouver les bonnes données. Nous nous inspirerons également des démarches suivies pour avoir une meilleure qualité d'image (changements de contraste etc).

II. Démarche projet

1. Méthode de gestion de projet utilisé

Afin d'atteindre nos objectifs dans le cadre de notre projet, nous avons mis en place des réunions hebdomadaires pour discuter de notre progression en tant qu'équipe. Ces réunions sont l'occasion de faire des démonstrations de nos travaux, d'identifier les points de blocage et de discuter des solutions pour les résoudre. Nous évaluons également si nous avons atteint nos objectifs initiaux et fixons de nouveaux objectifs si nécessaire.

De plus, si nous rencontrons des difficultés qui nécessitent une intervention de notre encadrant, nous planifions une réunion supplémentaire avec lui pour discuter de la situation et obtenir ses conseils et ses recommandations pour surmonter les obstacles.

Cette approche de réunion régulière et de suivi avec notre encadrant nous permet de maintenir une communication ouverte et transparente, d'identifier rapidement les problèmes et de les résoudre efficacement, tout en garantissant que notre projet reste sur la bonne voie pour atteindre nos objectifs.

Pour assurer une gestion de version efficace de notre projet, nous avons opté pour l'utilisation de Github. Nous avons créé un repository général où chaque module est organisé en sous-répertoires distincts. Ainsi, toutes les versions du projet sont stockées au même endroit, ce qui nous permet de suivre facilement les modifications apportées à chaque module au fil du temps.

En outre, nous avons mis en place un système d'issues sur Github, qui nous permet de signaler rapidement les bugs rencontrés ou les fonctionnalités à implémenter. Ce système facilite également la collaboration au sein de l'équipe, car il permet à chaque membre de suivre les problèmes en cours et les actions à entreprendre pour les résoudre.

En somme, l'utilisation de Github et du système d'issues contribue grandement à la gestion de version et au suivi efficace de notre projet, en permettant une communication transparente et une résolution rapide des problèmes rencontrés.

2. Rôles et responsabilités

Lors du lancement du projet, nous avons assigné chacun des 4 membres à une tâche spécifique pour l'initialisation du projet.

Jihene : Développement de l'application mobile

Saad : Développement de l'api ocr

Yessine : Développement de l'api pour le site web (backend)

Hugo : Développement de l'interface graphique du site web (frontend)

En fonction de l'avancement du projet, les membres qui ont progressé plus rapidement ont été affectés à d'autres tâches. Un tableau récapitulatif du travail effectué par chaque membre est disponible ci-dessous.

Partie	Tâche	Jihene	Saad	Yessine	Hugo
Analyse	Cas d'utilisation	x			
	Architecture	x	x	x	x
	Reprise de l'existant		x		
	Recherches	x	x	x	x
API OCR	Initialisation		x		
	Récupération des informations		x		
	JSON de retour avec les informations retrouvées		x		
	Affichage des infos récupérées sur l'image de la facture		x		
Mobile	Initialisation	x		x	
	Scan d'une facture et envoi vers l'API OCR	x		x	
	Affichage du retour du OCR	x		x	
	Authentification		x		
	Formulaire facture et envoi résultat vers API site web	x	x	x	
	Optimisations et correction bugs		x		
	Refonte de l'architecture	x		x	
Backend site	Initialisation			x	

web	Création de la base de données			x	
	Requête pour chaque objet (GET,POST,PUT,DELETE)			x	
	Requête login/logout			x	
	Requête mot de passe oublié et envoi mail			x	
	Gestion des rôles et des services			x	
Frontend site web	Initialisation				x
	Gestion de la création d'un compte/service				x
	Gestion du login / logout				x
	Gestion du "mot de passe oublié"				x
	Recherches des factures pour les services où l'on est gestionnaire				x
	Modification/suppression des utilisateurs/factures (si autorisé)	x		x	x
	Recherches des utilisateurs dont on est gestionnaire				x
	Création d'un super utilisateur		x	x	
	Blocage de l'accès si non gestionnaire				x
	Génération de note de frais		x		x
Rendu final	Rapport	x	x	x	x
	Déploiement (pas abouti)		x		
	Présentation	x	x	x	x

3. Outils Utilisés

Visual Studio Code

VS code est l'éditeur par excellence pour coder dans une multitude de langages.

Nous l'avons utilisé pour gérer nos applications javascript (front et back du site web) et python (api ocr).

Android Studio

Android Studio est l'IDE officiel pour coder une application android en Java ou Kotlin. Il y a d'autres choix sur le marché qui passent par des plugins pour supporter le développement mobile sur leur IDE (ex: intellij). Cependant, nous avons codé avant avec Android Studio pour du mobile c'est pour cela que nous le privilégions.

Nous l'avons donc choisi pour coder notre application mobile en Kotlin.

Django

Django est un framework de haut niveau permettant de développer des applications web en Python. Nous l'avons utilisé pour pouvoir exposer une API permettant d'utiliser notre OCR développé avec ce langage.

Django reste un framework assez conséquent comparé à quelque chose de plus léger comme Flask, cependant Flask mène facilement à certaines erreurs de connexion du côté de l'application mobile que nous n'avons pas réussi à résoudre (nous avons testé Django et Flask), ce qui nous a fait privilégier Django pour la version final de l'application.

NodeJS

Nous avons utilisé NodeJS pour notre serveur car il permet de facilement mettre en place une API, lancer des scripts et est rapidement déployable en ligne.

De plus, il possède "passport" qui permet facilement de mettre en place un système d'authentification sécurisé par des tokens et "mongoose", un ORM qui permet une mise en relation simple de notre backend et la base mongoDB.

React

React est un framework permettant de créer des interfaces web puissantes et mono page. Il met à disposition un système de vues et composants qui diminue la répétition de code et permet l'utilisation de composants créés par la communauté.

De plus, nous étions déjà familiers avec le framework grâce à un projet que nous avons terminé en décembre, ce qui nous a facilité le choix et nous utilisons donc React pour le frontend de notre site web.

4. Coût du projet

Nous n'avons pas été embauchés ni avons effectué le travail pour un professionnel, cependant nous pouvons simuler des coûts si cela avait été le cas.

Locaux

Nous avons travaillé au sein de l'université. Si l'on compare cela à une location, les locaux auraient coûté un minimum de 700 euros le mois, compte tenu que le travail a débuté en

janvier et prend fin en mars alors les locaux auront coûté environ 2800 euros pour les 4 mois.

Salaires

Nous avons travaillé 100h sur ce projet par personne, ce qui fait un total de 400h.

Nous pouvons utiliser comme référence un salaire de stagiaire d'environ 1000 euros par mois soit 1000 euros pour 151h, nous en calculons alors que pour 400h on coûtera $1000/151 \times 400 = \sim 2650$ euros.

Ajoutons à ce calcul le nombre d'heures que notre encadrant a passé à nous aider.

Si l'on suppose qu'il touche un salaire de 3500 euros pour 151h et qu'il nous a aidé 1h par semaine depuis le début du TPI (4h par mois donc 16 heures), nous en calculons donc que l'université aura dépensé $3500/151 \times 16 = \sim 370$ euros sur ce projet.

Tableau récapitulatif

Source	Coût
Locaux	2800 euros
Salaires	2650 euros
Coût d'encadrement	370 euros
Total	5820 euros

5. Risques du projet

Risque	Contraintes	Solution
Ne pas pouvoir rendre le projet à temps	Manque de temps nécessaire pour la réalisation du projet	Heures supplémentaires (non rémunérées...)
	Manque d'implication de tous les membres de l'équipe	Signalisation des membres de l'équipe problématiques aux encadrants
Ne pas pouvoir déployer les applications livrées	Manque de plateforme de déploiement	Avoir un serveur performant qui héberge les applications

		(surtout l'ocr qui nécessite de beaucoup de puissance de calcul)
Les applications livrées ne sont pas satisfaisantes	Les applications ne plaisent pas aux clients	Valider avec le client chaque fonctionnalité livrée pour ne pas avoir de surprise au dernier moment
	Les applications livrées ne sont pas utiles et restent inutilisées	Être certains par des tests utilisateurs que les applications sont faciles et agréables d'utilisation
	L'ocr est trop approximatif pour être utilisable	Se former plus sur les technologies associées à l'ocr qui sont hors notre champs d'expertise

III. Exigences fonctionnelles

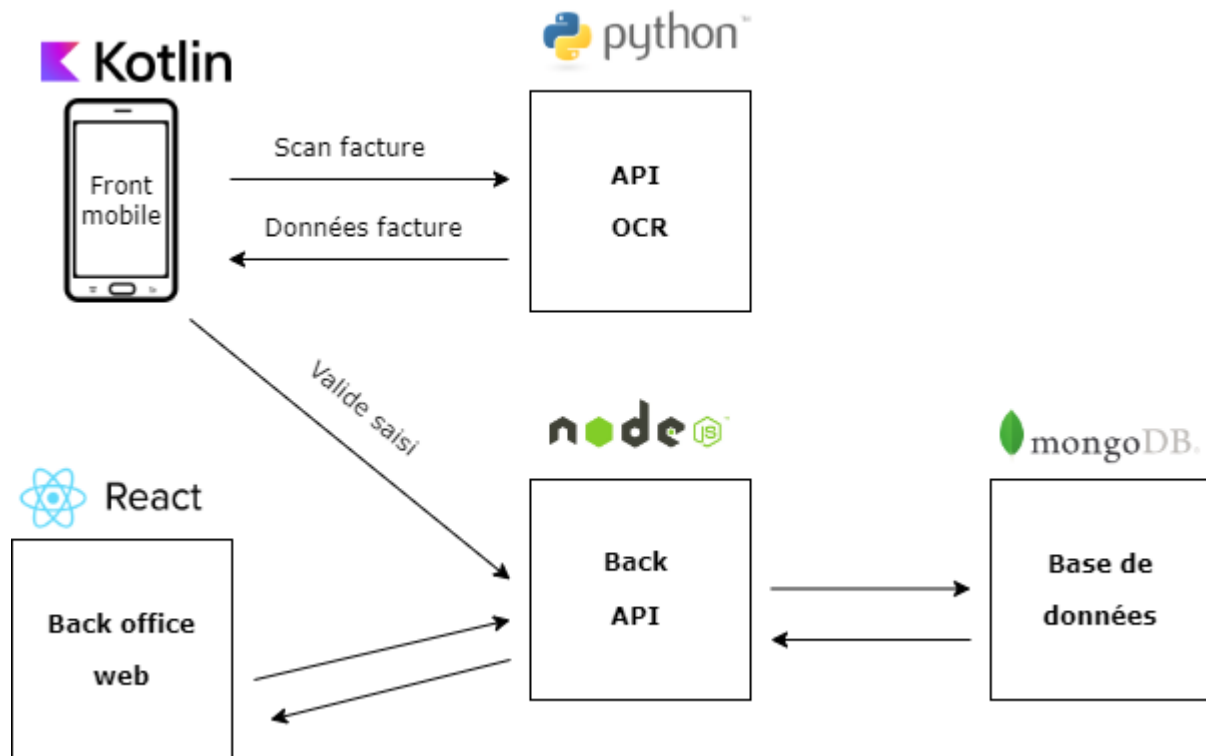
Suite à l'étude des besoins de l'application, nous avons pu en tirer (dans un premier temps) les cas d'utilisation suivants :

- Créer un compte utilisateur web (admin)
- S'authentifier (mobile/web)
- Attribuer un justificatif à un utilisateur: L'admin peut se connecter sur l'application mobile, scanner un justificatif et l'attribuer à un autre utilisateur.
- L'utilisateur web peut gérer son compte (supprimer son compte et modifier ses données personnelles)
- L'utilisateur mobile peut modifier son mot de passe
- L'utilisateur mobile peut scanner ses justificatifs
- L'utilisateur mobile peut consulter l'historique de ses justificatifs ajoutées
- L'utilisateur mobile peut valider le saisi des informations après l'analyse de l'OCR
- L'utilisateur web peut lister les utilisateurs inscrits
- L'utilisateur web peut gérer les utilisateurs (ajouter, modifier et supprimer un utilisateur)
- L'utilisateur web peut consulter les justificatifs conservés (par exemple pour une personne donnée ou pour un mois donné)
- L'utilisateur web peut modifier les données d'une facture donnée
- L'utilisateur web peut supprimer une facture donnée

- L'utilisateur web peut générer des notes de frais (une version pdf) à partir des justificatifs ajoutés par des utilisateurs

IV. Architecture système

Architecture logicielle



L'application mobile sera créée en Kotlin et permettra aux utilisateurs de se connecter et de scanner des factures. Ces factures seront envoyées à une API développée en Python avec le framework Django, qui effectuera une extraction des données nécessaires grâce à l'OCR et renverra les données extraites à l'application mobile.

L'utilisateur pourra ensuite corriger les données reçues et les valider, cette validation fera appel à un autre backend développé en Node.js qui gérera les données enregistrées dans la base de données MongoDB.

En outre, il y aura un back-office développé en React qui permettra aux gestionnaires de gérer les utilisateurs et les justificatifs présents dans la base de données.

V. Réalisations

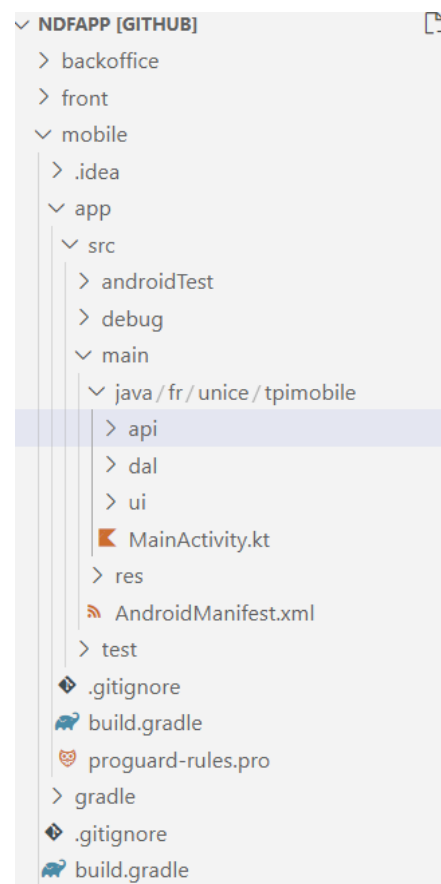
1. Application mobile

Description

L'application mobile permettra à l'utilisateur de se connecter et de scanner des factures, qui seront ensuite envoyées à l'OCR. Elle permettra également de récupérer les données extraites par l'OCR afin que l'utilisateur puisse les valider et les envoyer au backend pour qu'elles soient stockées dans notre base de données.

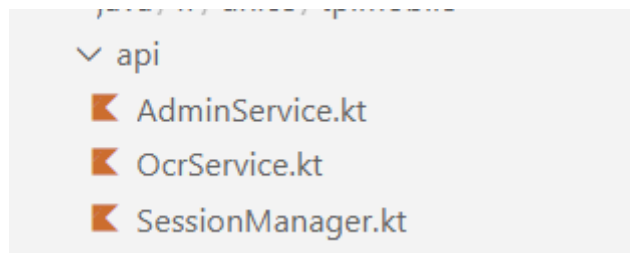
Conception du logiciel développé

Afin d'optimiser l'architecture de notre projet, nous avons cherché à la découper en différentes parties distinctes. Pour ce faire, nous avons proposé l'architecture présentée dans la figure ci-dessus.

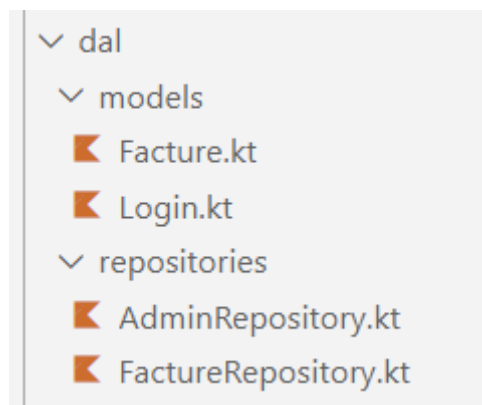


- La couche **api** inclut divers services, notamment le service "AdminService". Ce dernier contient des requêtes POST permettant à l'utilisateur de se connecter, ainsi qu'une autre requête pour créer une facture et envoyer les données de la facture numérisée par

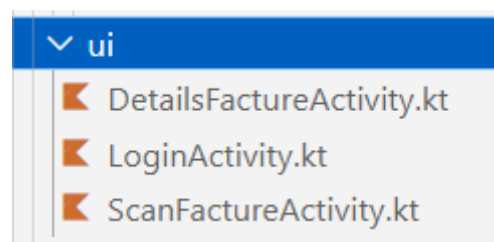
l'utilisateur à l'API backend pour les stocker dans notre base de données. Pour utiliser cette dernière fonctionnalité, il est probablement nécessaire d'être connecté en tant qu'utilisateur. Nous avons également un autre service "OcrService" qui permet d'envoyer la facture scannée à l'API de l'OCR via une requête POST. Cette requête renvoie un JSON contenant les informations pertinentes extraites par l'OCR. Nous trouvons également la classe SessionManager qui permet d'obtenir des informations sur la session de l'utilisateur connecté, facilitant ainsi la gestion des sessions utilisateur dans notre application.



- Les modèles Facture et Login ainsi que les repositories pour l'Admin et la Facture sont inclus dans la couche dal.



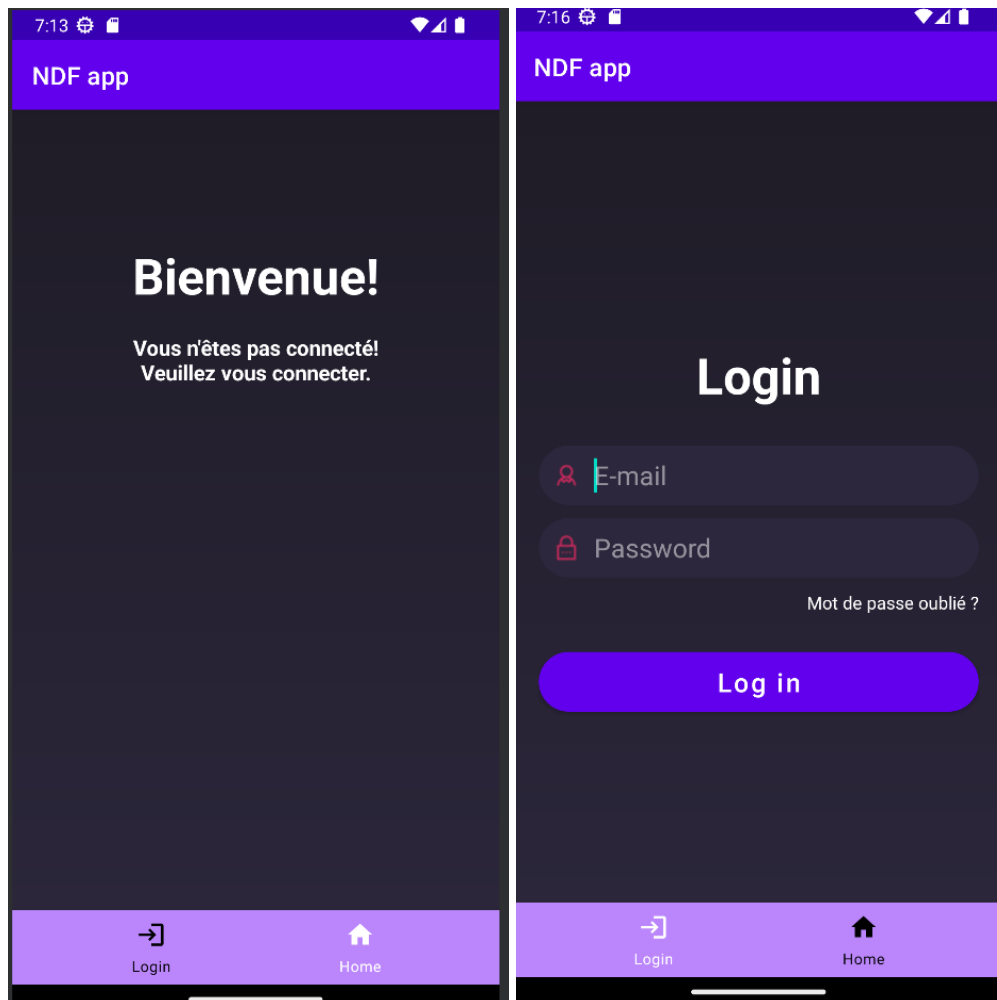
- Nous avons également la couche ui pour les différentes pages



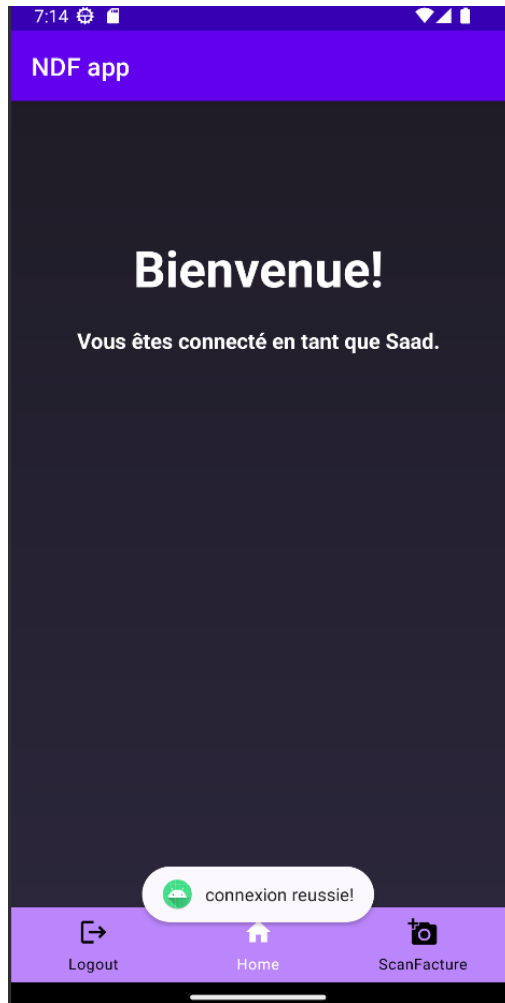
Réalisation

Lorsque l'application est lancée, la classe MainActivity est activée pour afficher la page d'accueil. Cependant, pour accéder à toutes les fonctionnalités de l'application, l'utilisateur doit se connecter en naviguant vers l'onglet login qui redirige vers la classe "LoginActivity".

Une fois que l'utilisateur a entré ses identifiants, une requête `@POST("login/")` du "AdminService" est envoyée à l'API du backend pour se connecter.



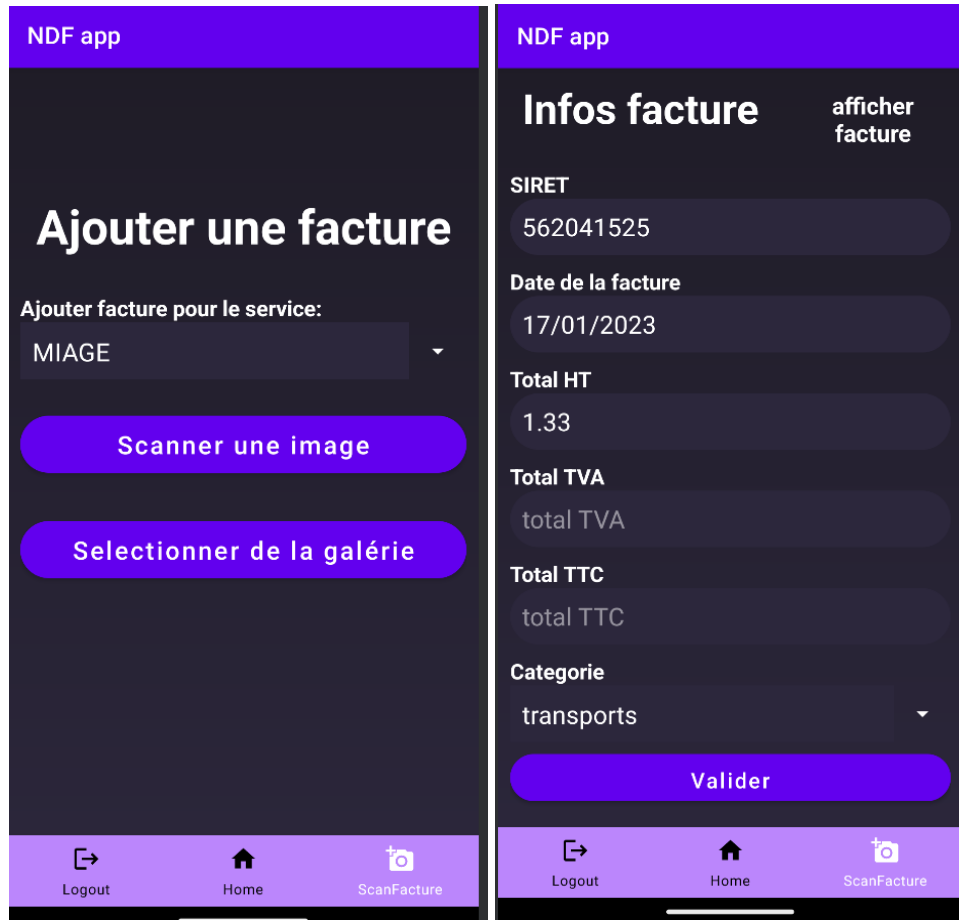
Si la connexion réussit, un message "Toast" apparaît sur la page d'accueil indiquant que la connexion a réussi et le nom d'utilisateur de l'utilisateur connecté est affiché.



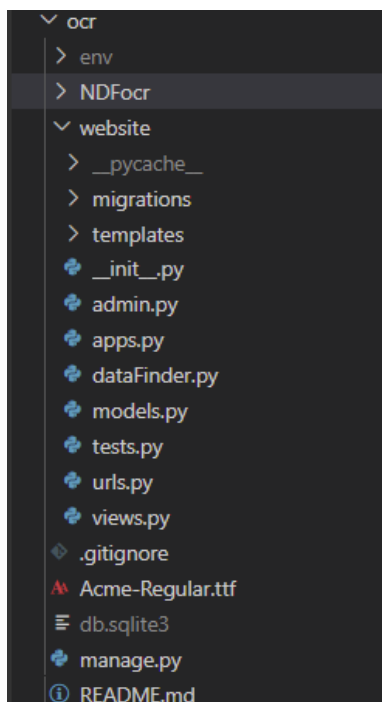
Dès que la connexion est établie, deux onglets sont disponibles pour l'utilisateur. Le premier permet à l'utilisateur de se déconnecter, tandis que le deuxième onglet, "ScanFactureActivity", permet à l'utilisateur de scanner une facture ou de sélectionner une facture prise en photo dans sa galerie.

Une fois que l'utilisateur a validé la photo, une requête `@POST("ocr/")` de "OCRService" est envoyée à l'API de l'OCR pour lire la facture et extraire les informations nécessaires sous forme de JSON, renvoyées à l'application mobile.

L'interface "DetailsFactureActivity" s'affiche ensuite avec un formulaire contenant les détails de la facture extraits par l'OCR (grâce au json envoyé). L'utilisateur peut modifier les informations si nécessaire et les valider. Lorsque l'utilisateur valide les informations, une requête `@POST("/api/factures")` est envoyée à l'api du backend pour insérer la facture dans la base de données. Notons que l'utilisateur peut également afficher la facture scannée.



2. API OCR



L'OCR a été codée avec le framework Django qui crée beaucoup de scripts dans l'arborescence pour démarrer le projet.

Nous allons nous focaliser que sur quelques fichiers en particulier :

- **dataFinder.py** : le script qui effectue une partie des manipulations nécessaires à partir des données reçues par l'ocr pour retrouver les données clé demandées.
- **views.py** : le script qui contient une définition des vues que nous allons mettre en place dans notre api. Ici on utilisera qu'une seule route "ocr/" qui accepte que des POST.

Nous allons expliquer le fonctionnement de l'api en détaillant étape par étape le traitement d'une requête reçue.

Traitement initial de la requête

A la réception de la requête, nous vérifions tout d'abord que c'est bien une requête POST, ensuite nous vérifions que le corps de la requête possède un fichier dont l'id est "img" avec l'image de la facture à traiter sous forme de stream de bytes.

Si ces vérifications échouent alors la requête renvoie une erreur 404, sinon nous passerons à la suite où la bibliothèque OCR rentre en jeu.

Nous avons utilisé easyOCR, cette bibliothèque va scanner la facture et nous renvoyer un tableau avec tous les éléments trouvés. Là où easyOCR se démarque par rapport aux autres bibliothèques, c'est qu'on aura la possibilité de connaître les coordonnées exactes de chaque élément chaque élément trouvé sur l'image sous forme de boîte englobante (par exemple le texte "machin" sera aux coordonnées $[[x,y],[x1,y1],[x2,y2],[x3,y3]]$ sur l'image, les coordonnées correspondant aux 4 côtés de la boîte englobante).

Avec la liste de tout le texte qui a été détecté et leur position, nous pouvons rentrer dans l'algorithme de détection qui se chargera de retrouver chaque élément clé.

Nous pouvons diviser la détection d'éléments clé en 2 :

1. La première catégorie vérifie directement si chaque texte trouvé sur l'image correspond à ce qui est attendu, dans cette catégorie nous retrouvons :
 - **date facture** : pour détecter la date facture nous vérifions si les bouts de texte correspondent aux formats jj-mm-yy, jj-mm-yyyy, jj/mm/yyyy, jj/mm/yy ou jj mois yyyy (ex : 11 avril 2023).
 - **catégorie** : nous allons associer une catégorie à la facture (transports, restauration, commerce, hébergement) en vérifiant si les bouts de texte correspondant aux mots clé d'une catégorie se trouvent sur la facture (ex : mention de "restaurant" sur la facture indique que c'est dans la catégorie "restauration").
2. La deuxième catégorie consiste à trouver des mots clé dans les bouts de texte et retrouver la donnée dans les bouts de texte adjacents :
 - **siret/siren** : nous détectons d'abord que la mention SIREN ou SIRET apparaît sur la facture, ensuite une fois cela trouvé, on concatène tous les bouts de texte qui suivent la mention jusqu'à avoir 9 ou 14 chiffres.
 - **tva/ttc/ht** : nous allons trouver la mention de TVA/TTC/HT dans le texte puis nous allons essayer de trouver le chiffre directement à droite de la mention ou en dessous. Nous allons effectuer cette opération à chaque fois qu'il y a la mention pour que s'il y a plusieurs mentions on prendra que le dernier qui

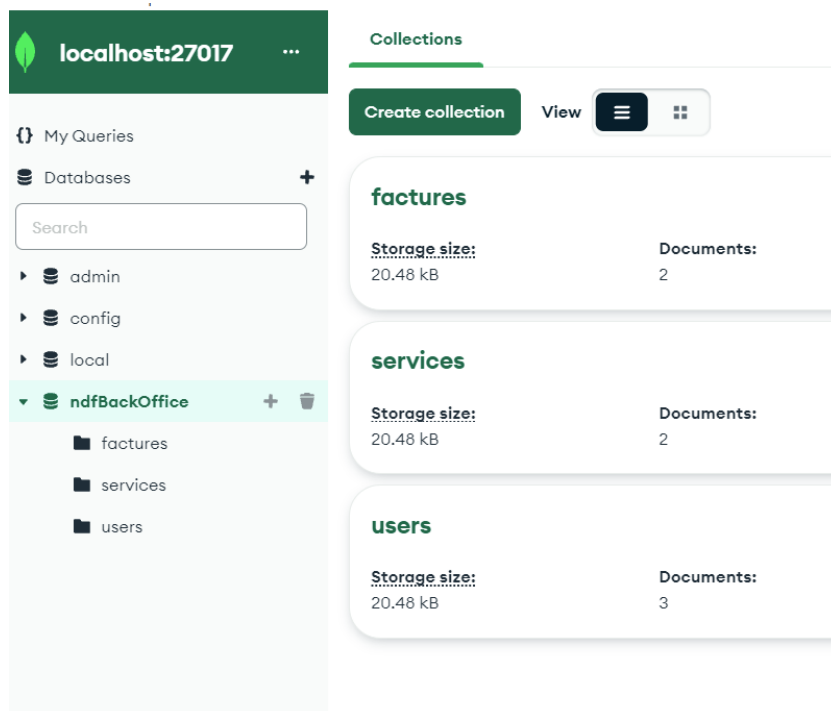
sera en principe un total. Si une des valeurs n'a pas été trouvée mais les autres si (ex : tva et ttc ont été trouvées mais pas ht) nous la calculerons manuellement.

Après avoir traité la facture et retrouvé les données clés, nous allons dessiner l'emplacement des éléments trouvés sur l'image de la facture que nous allons ajouter sous forme de base64 à la réponse.

3. Site web

Backend

Le backend de notre projet utilise la bibliothèque Mongoose pour permettre la communication avec la base de données, ainsi nous pouvons initialiser celle-ci en créant les tables nécessaires. Nous avons trois tables dans notre projet : "users", "factures" et "services".



Les utilisateurs possèdent un ou plusieurs services, pour chaque service, ils ont un rôle qui va déterminer leur droit (voir l'image ci-dessous).

ndfBackOffice.users

Documents Aggregations Schema Explain Plan Indexes Validation

Filter ⓘ ⓘ Type a query: { field: 'value' }

ADD DATA EXPORT COLLECTION

```

_id: ObjectId('64181b7d876cbf3468f0ac85')
prenom: "Yessine"
nom: "BEN EL BEY"
username: "Yessine"
superUtilisateur: false
mail: "yessine.benelbey@etu.unice.fr"
password: "$2b$10$5H0E.t2A04fx09xcfg/gbe1nabqpmFcNjd6yK7Lp40jmNIi04e2"
nomVoie: "Henri sappia"
numVoie: "84"
complementVoie: "Bat 7 esc 4"
codePostal: "06100"
commune: "Nice"
iban: "FR76 4567 8790 8798 789056781 15//"
bic: "15 //"
resetPasswordToken: "$2b$10$5H0E.t2A04fx09xcfg/gbe1nabqpmFcNjd6yK7Lp40jmNIi04e2"
services: Array
  0: Object
    _id: ObjectId('64181b7d876cbf3468f0ac86')
    role: "utilisateur"
    nom: "MIAGE"
    resetPasswordExpires: "1679301501842"
    __v: 0
  
```

Table users

La table facture aura tous les éléments nécessaires d'une facture pour générer une note de frais. Le champ créateur désigne le SIRET de l'entreprise.

```

_id: ObjectId('641eba449b970f2c100b8d07')
tva: 1.4
totalTtc: 2
totalHt: 1.33
créateur: "562041525"
dateAchat: 2023-02-17T13:00:00.000+00:00
idUtilisateur: "saad.ahmed@etu.unice.fr"
service: "MIAGE"
catégorie: "frais de déplacement"
__v: 0

```

```

_id: ObjectId('641eba449b970f2c100b8d06')
tva: 1.4
totalTtc: 2
totalHt: 1.33
créateur: "562041525"
dateAchat: 2023-01-17T13:00:00.000+00:00
idUtilisateur: "yessine.benelbey@etu.unice.fr"
service: "MIAGE"
catégorie: "transports"
__v: 0

```

Table factures

La table "service" contiendra les différents services proposés, ce qui nous permettra d'avoir accès à une liste de ces services en cas de besoin. Le champ "service" représente le nom du service.

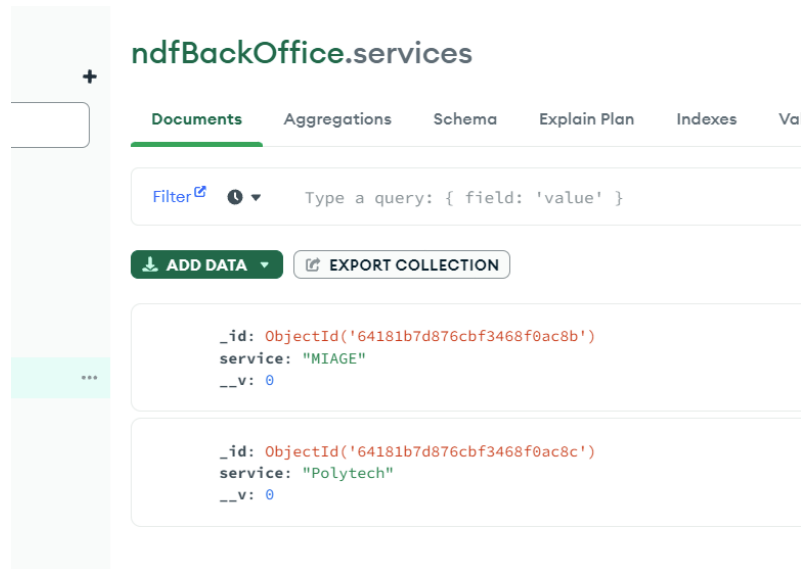


Table services

Pour chacune de ces tables, nous avons implémenté les requêtes de base telles que GET, POST, PATCH et DELETE.

En outre, nous avons des requêtes liées à la gestion de l'authentification. Nous utilisons "passport" pour créer des comptes, se connecter et se déconnecter. Il est nécessaire d'indiquer à "passport" que l'adresse email sera utilisée comme identifiant unique, ainsi qu'un mot de passe qui sera haché avant d'être stocké dans la table utilisateur.

Lors de la connexion, pour vérifier si l'utilisateur a saisi le bon mot de passe, nous allons hacher le mot de passe fourni et le comparer à celui stocké dans la base de données.

Il se peut que l'utilisateur oublie son mot de passe et donc il faut prévoir une requête permettant de changer son mot de passe. En ajoutant cette fonctionnalité nous nous sommes rendu compte qu'il allait faire des changements en base de données. Étant donné que l'utilisateur a oublié son mot de passe, il fallait lui donner un token provisoire par mail pour s'assurer de son identité. Le mail est envoyé grâce au module nodemailer et ressemblera à ceci:



tpagroupe2@gmail.com

À : Yessine Ben El Bey



Sam 25/03/2023 12:59

Vous recevez cet email car vous (ou une autre personne) avez demandé la réinitialisation du mot de passe pour votre compte.

Veuillez cliquer sur le lien suivant ou le copier-coller dans votre navigateur pour terminer le processus:

<http://127.0.0.1:3000/resetpassword/?token=50b985ed7644e494ebeb3e0e0529af4b516bbfa>

Si vous n'avez pas effectué cette demande, veuillez ignorer cet email et votre mot de passe restera inchangé.

Email permettant de changer son mot de passe

Ce token provisoire va donc être un attribut de la table user qui sera nommé "resetPasswordToken". Par défaut cet attribut à la même valeur que le mot de passe haché, et dès que nous faisons une demande de réinitialisation de mot de passe il va changer. De plus, il est nécessaire d'assurer la validité du token pendant une durée spécifique. Pour ce faire, nous avons introduit un nouvel attribut appelé "resetPasswordExpires" dans la table utilisateur, qui permet de déterminer si le token a expiré ou non. Nous veillons à ce que le token expire au bout d'une heure. Lorsqu'une demande de réinitialisation de mot de passe est effectuée, nous enregistrons la date correspondante en millisecondes sous le nom de "resetPasswordExpires", puis la comparons avec la date actuelle à laquelle nous ajoutons 3600000 millisecondes, soit une heure.

Après avoir cliqué sur le lien reçu par email, vous serez redirigé vers l'interface vous permettant de changer votre mot de passe. Une fois le mot de passe modifié, vous recevrez un email de confirmation de la modification.



tpagroupe2@gmail.com

À : Yessine Ben El Bey



Sam 25/03/2023 13:32

Ceci est une confirmation que le mot de passe pour votre compte "yessine.ben-el-bey@etu.unice.fr" a été modifié.

Je confirme.

Voici le mot de passe.

Cela ne marche pas.

Email de confirmation de changement de mot de passe

Après avoir présenté le back-end du site web, nous allons maintenant vous présenter le front-end.

Frontend

C'est ReactJS qui a été choisi.

Cette solution est, tout d'abord, une solution que nous connaissons bien car nous l'avons utilisée précédemment mais aussi dans un second temps une solution qui s'avère la plus efficace, facile et viable et ce notamment grâce au Hook « useState ».

D'après la documentation officielle de React, un Hook est une fonction qui permet de « se brancher » sur des fonctionnalités React.

Par exemple, useState est un Hook qui permet d'ajouter l'état local React à des fonctions composantes. Appeler useState déclare une « variable d'état ».

C'est un moyen de préserver des valeurs entre différents appels de fonctions.

Normalement, les variables disparaissent quand la fonction s'achève mais les variables d'état sont préservées par React.

Le seul argument à passer au Hook useState() est l'état initial. useState renvoie une paire de valeurs : l'état actuel et une fonction pour le modifier.

```
import React, { useState } from 'react';

function Example() {
  // Déclare une nouvelle variable d'état, que nous appellerons « count »
  const [count, setCount] = useState(0);
```

Nous déclarons une variable d'état appelée count, et l'initialisons à 0. React se rappellera sa valeur entre deux affichages et fournira la plus récente à notre fonction. Si nous voulons modifier la valeur de count, nous pouvons appeler setCount.

On peut alors écrire le paragraphe suivant : <p>Vous avez cliqué {count} fois</p>

Le front-end sera conçu avec la bibliothèque [chakra-ui](#) que nous connaissons bien et que nous apprécions beaucoup. Les validations des champs seront faites avec la bibliothèque [validator](#) et les requêtes vers la base de données MongoDB seront faites à partir d'[axios](#).

L'application possède un thème clair ainsi qu'un thème sombre paramétrable dans la barre de navigation (navbar).

Cette navbar permet également de se connecter / se déconnecter et d'effectuer toutes les tâches que peut faire un gestionnaire si et seulement si l'utilisateur connecté est un gestionnaire d'un service.

Être gestionnaire d'un service signifie que l'on peut gérer les utilisateurs et les factures liées à ce même service.

Page de connexion

L'application propose donc une page de connexion :

[Accueil](#)[Se connecter](#)

Se connecter

Adresse courriel *

Mot de passe *

[Mot de passe oublié ?](#)

Valider

© 2023 TPI NDF. Agrebi, Ahmed, Ben El Bey, Nortier

Page mot de passe oublié

Une page si l'on a oublié son mot de passe:

[Accueil](#)[Se connecter](#)

Mot de passe oublié

 Un courriel de réinitialisation de mot de passe vous a été envoyé à l'adresse hugo.nortier@free.fr

Adresse courriel *

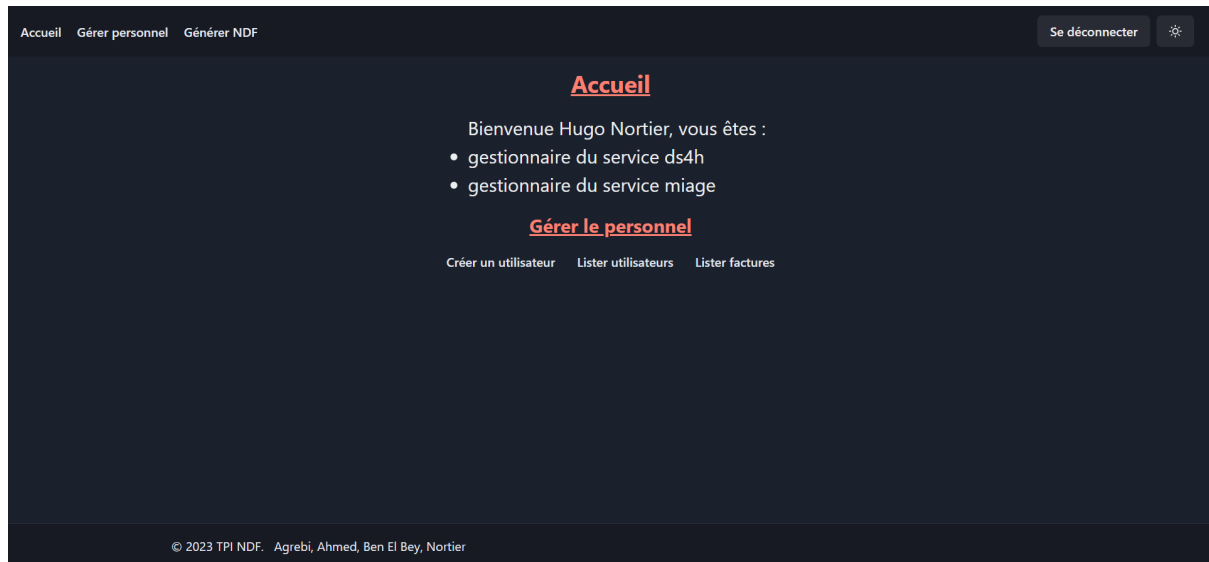
Recevoir courriel de réinitialisation

© 2023 TPI NDF. Agrebi, Ahmed, Ben El Bey, Nortier

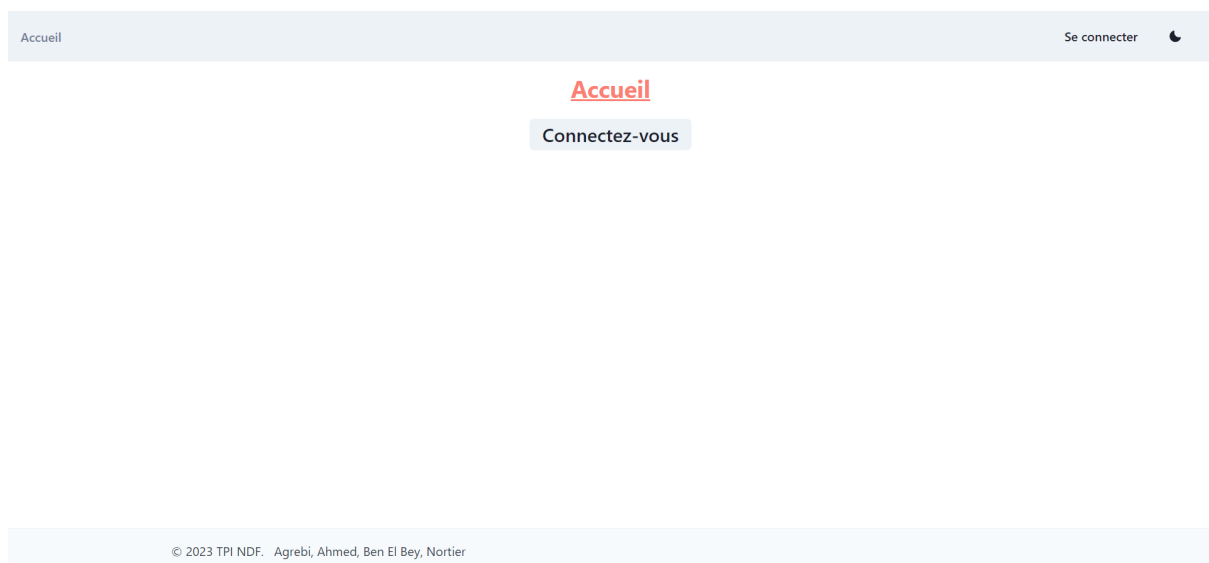
Puis un courrier électronique est envoyé et l'on peut modifier le mot de passe en suivant le lien à l'intérieur de l'e-mail.

Page d'accueil

Lorsque l'on est connecté en tant que gestionnaire, la navbar affiche la possibilité d'effectuer des tâches de gestionnaire et la page d'accueil présente notre profil. (ici vous apercevrez également le mode sombre) :



Un utilisateur non connecté sera invité à se connecter.



Un superUtilisateur verra, quant à lui, des choses différentes.

En effet, il se verra le droit de créer un nouveau service et seul lui peut le faire :

Accueil

Bienvenue Hugo Nortier, vous êtes
superUtilisateur:

Gérer le personnel

Créer un utilisateur Créer un service Lister utilisateurs Lister factures

Page de création d'utilisateur

Alors, il est possible de créer un utilisateur. Pour cela, un formulaire Multi-Step a été utilisé. Un formulaire Multi-Step est un formulaire par étapes où l'utilisateur peut naviguer avec des boutons « suivant » et « précédent » entre les étapes. Cela est plus ergonomique, moins indigeste et permet de ne pas avoir tous les champs de création au même endroit. Cela permet également de regrouper les champs par catégorie (e.g: l'étape 4 est celle dite "comptable" où sont renseignés l'IBAN et le BIC de l'utilisateur).

Il y a 4 étapes dans ce formulaire:

étape 1 : les données de l'utilisateur

(Le mot de passe est stocké hashé dans la base de données pour la sécurité).

S'il y a une erreur lors de la saisie, on empêche l'utilisateur de se rendre à l'étape suivante, on colore en rouge la cellule de saisie et on affiche l'erreur.

Dans les figures qui suivent, le mot de passe est, dans un premier temps, trop peu complexe et ensuite il ne correspond pas à la confirmation (second champ de saisie du mot de passe). Enfin, l'IBAN de l'utilisateur n'est pas conforme. En effet, la librairie "validator" est très complète et propose des validations d'IBAN, d'e-mail, de BIC, de carte bancaire, de code postal, de numéro de téléphone etc et ce pour n'importe quel pays.

Créer un compte

Adresse courriel *

Prénom *

Nom *

Mot de passe *

Mot de passe trop peu complexe
8 caractères (1 majuscule, 1 minuscule, 1 spécial)

Confirmation mot de passe *

Suivant →

Créer un compte

Adresse courriel *

Prénom *

Nom *

Mot de passe *

8 caractères (1 majuscule, 1 minuscule, 1 spécial)

Confirmation mot de passe *

Ces mots de passe ne correspondent pas

Suivant →

Créer un compte

IBAN *

International Bank Account Number erroné

Codes BIC / SWIFT *

← Précédent

Valider

étape 2 : les services et les rôles de l'utilisateur.

Créer un compte

Associer un nouveau service

Service n°1 *

Accréditation du service n°1 *

☐ utilisateur ☒ gestionnaire

Service n°2 *

Accréditation du service n°2 *

☒ utilisateur ☐ gestionnaire

← Précédent Suivant →

La personne qui crée le compte utilisateur peut associer un rôle et un service au nouvel utilisateur. Néanmoins, il ne peut pas faire cela si lui-même n'est pas gestionnaire dans le service en question.

(N.B: les super-utilisateurs contournent cette règle).

Les services listés sont ceux ayant été créés en amont par le formulaire de création de service (voir plus bas).

La liste de rôles-services est dynamique. Appuyer sur "ajouter un nouveau service" redimensionner la liste.

étape 3 : l'adresse de l'utilisateur

Le complément est facultatif. Le code postal est validé par la librairie validator qui vérifie que ce soit un code postal au format français.

étape 4 : les informations bancaires de l'utilisateur :

L'IBAN et le code BIC sont renseignés dans cette étape.

Créer un compte

IBAN *

FR763000100794123456789018

Codes BIC / SWIFT *

AGRIFRPP847

← Précédent





Valider

En cas de création réussie, un message (dont le design est semblable à celui visible lors de la présentation de l'écran de mot de passe oublié) nous l'informe sinon, l'erreur est également affichée.

Page de listing des utilisateurs

Les utilisateurs des services dont on est gestionnaire sont listés. On peut les supprimer.

Lister Utilisateurs

MAIL	NOM PRÉNOM	ADRESSE	IBAN	BIC	MODIFIER	SUPPRIMER
hugo3.nortier@orange.fr	Nortier Hugo	27 bis rue dupont, bat 7, 75000 versailles	FR7630001007941234567890185	AGRIFRPP847		
g2@g.fr	d Hugo	3 rue des potiers, 78600 versailles	FR7630001007941234567890185	AGRIFRPP847		
hugo.nortier@sfr.fr	Nortier Hugo	27 bis rue dupont, 06600 Antibes	FR7630001007941234567890185	AGRIFRPP847		

ou bien les modifier:

Page de modification d'un utilisateur

[Accueil](#) [Gérer personnel](#) [Générer NDF](#)

Utilisateur modifié.

[Se déconnecter](#)

Modifier l'utilisateur g2@g.fr

L'utilisateur doit modifier son mot de passe en faisant "mot de passe oublié" sur la page de connexion

Modifier user

User modifié avec succès

Prénom *

Hugo

Nom *

d

Numéro de voie *

3

Nom de voie *

rue des potiers

Complément de voie

complément de voie

Code postal *

78600

Commune *

versailles

IBAN *

FR7630001007941234567

Codes BIC / SWIFT *

AGRIFRPP847

Valider







© 2023 TPI NDF. Agrebi, Ahmed, Ben El Bey, Nortier

Page de listing des factures

Les factures des utilisateurs faisant partie des services dont on est gestionnaire sont listées. Tout comme pour les utilisateurs, on peut les supprimer.

Lister Factures

Filtrer par utilisateur : tous les utilisateurs Lister







EMAIL	DATE FACTURE	SERVICE	TOTAL HT	TOTAL TTC	TVA	CRÉDITEUR	CATÉGORIE	MODIFIER	SUPPRIMER
hugo3.nortier@orange.fr	2022-05-10	miage	21.25	25.5	4.25	carrefour antibes	alimentation		
hugo.nortier@free.fr	2022-05-10	miage	10.83	12.99	2.16	casino villeneuve-loubet	alimentation		
hugo1.nortier@sfr.fr	2022-05-10	miage	24.56	29.47	4.91	auchan grasse	alimentation		

Si l'on ne touche à rien, nous listons toutes les factures des services où nous sommes gestionnaires. Sinon, nous pouvons filtrer par utilisateur grâce au champ de saisie présent sur la page. Ici, nous filtrons les factures d'un user en particulier :

Accueil Gérer personnel Générer NDF Se déconnecter

Lister Factures

Filtrer par utilisateur : hugo3.nortier@orange.fr Lister

EMAIL	DATE FACTURE	SERVICE	TOTAL HT	TOTAL TTC	TVA	CRÉDITEUR	CATÉGORIE	MODIFIER	SUPPRIMER
hugo3.nortier@orange.fr	2022-05-10	miage	21.25	25.5	4.25	carrefour antibes	alimentation		
hugo3.nortier@orange.fr	2022-05-10	miage	10.83	12.99	2.16	casino villeneuve-loubet	alimentation		
hugo3.nortier@orange.fr	2022-05-10	miage	24.56	29.47	4.91	auchan grasse	alimentation		

© 2023 TPI NDF. Agrebi, Ahmed, Ben El Bey, Nortier

On peut également modifier une facture

Accueil Gérer personnel Générer NDF

Facture modifiée.

Se déconnecter

Modifier la facture 640f6dfb2674750d30a2033f

Modifier facture

Facture modifiée avec succès

Adresse courriel *

hugo1.nortier@orange.fr

Date achat *

10/05/2022

Service *

miage

Créateur *

auchan grasse

Total HT *

24.56

Total TTC *

29.47

TVA *

4.91

Catégorie *

alimentation

Valider

© 2023 TPI NDF. Agrebi, Ahmed, Ben El Bey, Nortier

Page de création d'un service

La création d'un service nécessite le fait d'être superUtilisateur.

La création s'effectue au travers d'une fenêtre "[modale](#)".

Les fenêtres modales sont des fenêtres qui apparaissent directement à l'intérieur de la fenêtre courante du navigateur, au-dessus de la page web qui les appelle comme on le voit ci-dessous.

Accueil Gérer personnel Générer NDF

Se déconnecter

Créer un service

Le service Polytech a été créé

Nom du service *

Polytech

Créer Quitter

© 2023 TPI NDF. Agrebi, Ahmed, Ben El Bey, Nortier

Page de création d'une note de frais

Enfin, une note de frais est créée pour un utilisateur dont on est gestionnaire de son service et pour un service donné.

Pour créer cette note de frais, toutes les factures entre deux dates (une de début, une de fin) sont assemblées.

Par exemple, ci-dessous, nous souhaitons créer une note de frais pour Hugo3.Nortier@orange.fr concernant toutes ses factures comprises entre le 01/05/2022 et le 31/05/2022 pour le service Miage.

Malheureusement, il n'y a pas de factures comprises entre ces dates là.

N.B: si la date de fin est inférieure à la date de début, une erreur s'affiche.

En retentant et en prenant cette fois-ci des dates comprises entre le 01/03/2023 et le 31/03/2023, on obtient en résultat un pdf qui s'affiche en dessous.

Accueil
Gérer personnel
Générer NDF
Se déconnecter

Créer une Note de Frais

Pour l'utilisateur : *

hugo3.nortier@orange.fr

Pour le service : *

miage

entre (Date début) *

01/03/2023

et (Date fin) *

31/03/2023

Créer NDF

NOTE DE FRAIS – Service MIAGE

Hugo NORTIER
12 Ave. du balai, bat. 7
06600 Antibes

Date	Créancier	Total TTC	Total HT	TVA
01/03/2023	Carrefour ANTIBES	25.50	21.25	4.25
05/03/2023	Auchan GRASSE	12.99	10.83	2.16
12/03/2023	Carrefour NICE TNL	29.47	24.56	4.91
25/03/2023	Maxi Coffee	1.0	0.83	0.17

Total TTC : 68.96
Total HT : 57.47
TVA : 11.49

Précédent
Suivant

C'est la librairie "[react-pdf](#)" qui a permis de créer ce pdf et donc cette note de frais.

Conclusion

Notre projet de thèse d'innovation a porté sur l'automatisation des notes de frais, une tâche souvent fastidieuse pour les employés et les entreprises. Grâce à notre travail, nous avons réussi à développer une solution complète comprenant une application web, une application mobile et une technologie de reconnaissance optique de caractères (OCR).

Nous sommes fiers des réalisations que nous avons accomplies jusqu'à présent et sommes convaincus que cela représente un point prometteur pour la suite de notre projet. En effet, notre solution offre de nombreux avantages tels que la rapidité, la précision et l'élimination des erreurs humaines, ce qui permet aux entreprises de gagner du temps et de l'argent.

À court terme, l'objectif est de développer une nouvelle fonctionnalité pour notre application Android, qui permettra aux gestionnaires de numériser des factures et de les attribuer à un autre utilisateur. Nous envisageons également d'améliorer notre OCR afin qu'il puisse extraire plus d'informations et améliorer sa précision.

De plus, il est essentiel que nous déployions l'ensemble de notre solution et que nous ajoutons un fichier de configuration permettant de changer l'url du site web en fonction de si le projet est lancé en déploiement ou non.

En outre, il sera nécessaire de permettre la modification et la suppression des services.

À moyen terme, nous prévoyons d'ajouter une fonctionnalité permettant aux utilisateurs de notre application mobile de consulter l'historique des justificatifs qu'ils ont ajoutés. Nous avons également pour objectif d'améliorer notre application web en filtrant les utilisateurs par service plutôt que de lister tous ceux dont on est gestionnaire.

À plus long terme, nous envisageons d'améliorer la réactivité et le design de l'interface utilisateur (UI) de notre application web. De plus, nous souhaitons créer des services avec plus d'informations (photo, adresse du service, numéro de téléphone du service).

Il convient de noter qu'il y a eu des **problèmes** rencontrés au cours de notre projet.

Plusieurs obstacles ont entravé le déploiement de notre solution, en particulier le coût élevé associé à sa mise en place.

De plus, nous avons éprouvé des difficultés à effectuer les tests nécessaires sur Android, car nous n'avons pas réussi à utiliser le simulateur pour prendre des photos de factures.

Le manque de ressources a également posé un défi, notamment en raison de la lenteur de nos ordinateurs lorsque nous avons essayé de lancer l'ensemble du projet. Cela est dû à la consommation importante de mémoire, car notre solution intègre un backoffice, un OCR et une application mobile. De plus, nous n'avons pas été en mesure d'améliorer le taux de réussite de notre OCR, en raison d'une pénurie de temps et de compétences.

Ce projet nous a permis de développer nos compétences en travail d'équipe et en gestion de projet de manière méthodique et organisée. Nous avons eu l'opportunité de consolider nos connaissances en associant nos acquis théoriques à l'environnement pratique. Grâce à cette expérience, nous avons pu renforcer nos compétences techniques, notamment en ce qui concerne React, NodeJS, Android et Python. Nous avons également appris l'importance de la communication pour résoudre les problèmes rencontrés.

Références

Easy OCR, URL : <https://github.com/JaidedAI/EasyOCR>

Django, URL: <https://www.djangoproject.com/>

MongoDb, URL: <https://www.mongodb.com/>

Mongoose, URL: <https://mongoosejs.com/>

Docker, URL: <https://www.docker.com/>

React, URL: <https://react.dev/>

NodeJs, URL: <https://nodejs.org/en>

NodeMailer, URL: <https://nodemailer.com/about/>

Passport, URL: <http://www.passportjs.org/packages/passport-jwt/>

Retrofit, URL: <https://square.github.io/retrofit/>