



République Islamique de Mauritanie  
Ministre de l'Enseignement Supérieur et de la  
Recherche Scientifique



## MÉMOIRE DE STAGE DE FIN D'ÉTUDES

Présenté en Vue de l'obtention du  
Diplôme de Master Professional en Informatique  
Appliquée à la Gestion (MPIAG)

Par:

Mbare Mohamed Salem Belkher (IE19275)

DÉVELOPPEMENT D'UN APPLICATION MOBILE POUR  
TRÉTER LES REFERENCE DU CLIENT DU SOMELEC

Encadreur pédagogique et professionnel :

Dr.eng Mohamedou Cheikh tourad

Réalisé au sein de NextTechnology

**Next**

T E C H N O L O G Y

Année Universitaire 2022-2023

# Dédicace

- Je dédie ce travail:
- À mes chers parents, source de bonheur, d'espoir et de vie. Si je devais parler dès demain, je ne trouverais pas assez de mots pour exprimer tout le respect, les remerciements et la gratitude que j'ai envers vous. Vous m'avez soutenu dès le début et n'avez jamais désespéré. Au contraire, vous avez bravé toutes les difficultés pour faire de moi ce que je suis aujourd'hui.
- À mes proches amis, collègues et à ceux qui ont partagé avec moi tous les moments heureux et tristes tout au long de ce parcours.
- À tous les membres de ma famille et à mes amis.
- À tous mes formateurs et à toute l'équipe administrative de Next-Technology pour l'aide qu'ils ont toujours apportée aux stagiaires.
- À toute personne qui, de près ou de loin, a participé à ma formation.

# **Remerciements**

Je remercie avant tout Allah de nous avoir donné la volonté, la patience et la santé pour réaliser ce modeste travail. J'adresse mes remerciements les plus chaleureux à:

- Dr. Emany Mohamed SIDI, coordinateur de MPIAG, ainsi qu'à
- Dr.eng Mohamedou Cheikh Tourad, Directeur technique de Next-Technology et mon encadrant académique, pour sa disponibilité et ses conseils.

Je remercie également DG. Mouhmed Ould Hmeimed, le directeur général de la structure d'accueil, de m'avoir donné cette opportunité. Et pour avoir mis les moyens nécessaires au bon déroulement de ce stage de six mois et à la réalisation de ce travail.

Je tiens également à remercier tous mes professeurs qui m'ont enseigné durant deux ans de formation en master en Informatique de gestion, pour leurs précieux conseils et orientations.

# Abstract

This document presents the work carried out during my end-of-study internship to obtain a professional master's degree in computer science applied to management (**MPIAG**) at the Higher Institute of Accounting and Business Administration (**ISCAE**) and the University of Nouakchott Al Aasriya (**UNA**). It presents the realization of a project carried out during an internship within the company **Next-Technology**. This project is related to the quality Somelec

# Resumé

Ce document présente le travail réalisé lors de mon stage de fin d'études pour l'obtention d'un master professionnel en informatique appliquée à la gestion (MPIAG) à l'Institut Supérieur de Comptabilité et de Gestion des Entreprises (ISCAE) et à l'Université de Nouakchott Al Aasriya (UNA). Il présente la réalisation d'un projet réalisé lors d'un stage au sein de la société Next-Technology consistant à créer une Application Mobil de somelec pour ses références, abonnements, branchements et factures.

Cette solution offre aux entreprises les outils nécessaires pour mieux contrôler et suivre leurs flux de références, abonnements, branchements et factures grâce à une interface mobile, une couche administrative de traitement, une couche de stockage ainsi qu'un système SMS. L'intégration de ces composants permet une gestion plus efficace de l'application.

L'architecture de la solution s'appuie sur des technologies telles que Flutter, Django, Google Cloud, Django Rest Framework, PostgreSQL et Heroku, qui garantissent une mise en œuvre solide, évolutive et flexible.

# Table des matières

Chapitre 1 : Introduction .....	1
1.1 Contexte .....	1

1.2 Motivations .....	1
1.3 Problématiques.....	1
1.4 Objectifs .....	2
1.5 Organisation du manuscrit. ....	2
Chapitre2: Structure d'accueille .....	3
2.1 Contexte.....	3
2.2 Présentation de l'entreprise .....	3
2.3 Méthodologie du travail : .....	4
2.3 Choix d'une méthodologie de travail .....	4
2.4 Expérience de stage .....	7
Chapitre 3 : Expérience utilisateur .....	9
3.1 Introduction .....	9
3.2 Le processus de UX .....	9
1. Compréhension :.....	9
2. Recherche (Research):.....	9
3. Esquisser (Sketch):.....	10
4. Concevoir (Design):.....	10
5. Test:.....	10
Chapitre 4 : Environnement de travail .....	16
4.1 Infrastructure matérielle. ....	16
4.2 Système d'exploitation .....	16
4.3 Conception.....	16
4.3.1 Modelisation .....	16
4.3.2 Langage de modélisation .....	16
4.3.3 Logiciel de modélisation .....	16
4.6 Développement .....	18
4.6.1 Backend .....	18
4.6.1.1 Framework de backend .....	18
4.6.1.2 Système de gestion de base de données .....	18
4.6.2 Front-end .....	20
4.6.2.1 Motif d'architecture Front-end .....	20
4.6.3 Ide .....	21
4.6.4 Gestionnaires de dépendances .....	22

4.6.5 Gestionnaire des versions.....	22
4.7 Déploiement .....	23
4.8 Rédaction .....	24
4.8.1 Rapport .....	24
4.8.2 Présentation.....	24
Chapitre 5 : Conception de l'application .....	25
5.1 Rappel sur le besoin.....	25
4.4 Diagramme de classe .....	26
Chapitre 6 : Les soutils utilises et technologies .....	29
6.1 Python.....	29
6.2 Django .....	29
6.3 Dart .....	29
6.4 Flutter .....	29
6.5 vs code .....	30
6.6 Android studio .....	30
6.7 Postman .....	30
6.8 Figma.....	30
6.9 Start UML.....	31
6.10 Git.....	31
6.11 Bitbucket.....	31
Chapitre 7 : Le Réalisation de l'application .....	31
7.1 Backend.....	32
7.1.1 Base de données .....	32
7.1.2 Système d'envoie SMS by phone number .....	33
7.1.3 Operations CRUD .....	34
7.2 Frontend .....	35
Conclusion générale.....	47
Références .....	47

# Liste des figures

Figure 1 : le fonctionnement de la methode scrum .....	14
Figure 2 : scrum board .....	15
Figure 3 : composantes de mon design d'utilisation .....	20
Figure 4: page d'authentification .....	20
Figure 5: forget password .....	21
Figure 6: home (no ref) .....	22
Figure 7 : home(ref).....	22
Figure 8 : mes facture .....	23
Figure 9 : demandes .....	23
Figure 10: auto-releve .....	24
Figure 11: history .....	24
Figure 12 : profile .....	25
Figure 13 : Architecture MVC .....	28
Figure 14 : Architecture Cubit .....	31
Figure 15 : diagramme de cas d'utilisation client .....	36
Figure 16 : diagramme de cas d'utilisation adminstrateur .....	37
Figure 17 : diagramme de class .....	38
Figure 18: configuration GCP .....	45
Figure 19: on splash screen .....	50
Figure 20: on boarding screen.....	51
Figure 21: login screen.....	52
Figure 22: register screen.....	53
Figure 23: OTP .....	53
Figure 24: setup password .....	53
Figure 25: forget password .....	54

Figure 26: home screen (no ref) .....	55
Figure 27: home (ref) .....	56
Figure 28: facture .....	57
Figure 29: Auto releve .....	58
Figure 30: historique .....	59
Figure 31: profile screen .....	60

# Chapitre 1 : Introduction

## 1.1 Contexte

Dans le contexte des entreprises, la réception des factures et la vérification des documents déposés par l'utilisateur peuvent prendre du temps, même au niveau de l'utilisateur. Le fait de passer la journée à l'agence de la SOMELEC pour déposer les documents nécessaires pour un branchement ou un abonnement, selon sa nature, représente un défi pour la plupart des personnes. Pour résoudre cette problématique, nous avons proposé une application mobile qui gère tous ces défis de manière efficace.

## 1.2 Motivations

L'adoption d'une application efficace de gestion des factures de la SOMELEC est motivée par plusieurs facteurs. La société cherche à améliorer sa productivité en évitant les blocages dans le traitement et la vérification des documents, ainsi qu'en facilitant l'envoi des factures aux utilisateurs. De plus, elle souhaite automatiser et rationaliser ce processus afin de gagner du temps. De surcroît, notre application offre un service de paiement des factures.

## 1.3 Problématiques

La réalisation de ce système nécessite tout d'abord d'obtenir des réponses aux questions suivantes, qui ne sont pas exhaustives mais servent de point de départ pour aborder les problématiques liées à l'application au sein des entreprises :

- ❖ Comment s'assurer que les données de la Somelec soient confidentielles et non partageables ?

- ❖ Comment optimiser le stockage, l'organisation et l'accès aux factures pour faciliter leur consultation ?
- ❖ Comment optimiser le dépôt de documents pour éviter les blocages ? [OBJ]
- ❖ Comment garantir une satisfaction client élevée en offrant des délais de réponse rapides ?
- ❖ Comment proposer une solution pour la résolution des dépannages afin de répondre aux besoins des clients de Somelec ?
- ❖ Comment proposer à l'utilisateur une solution qui englobe leurs références tout en offrant la possibilité d'avoir les arriérés affectés à la référence correspondante ?

## **1.4 Objectifs**

En nous basant sur les problématiques mentionnées précédemment, l'objectif de ce projet de fin d'études (PFE) consiste donc à concevoir et réaliser une application qui représente la meilleure solution pour surmonter ces problèmes de manière efficace et optimale, tout en respectant la confidentialité des données des employés et de l'entreprise. Un objectif important de cette application est de faciliter aux clients de Somelec d'éviter les blocages et les retards dans le traitement des documents envoyés par l'utilisateur.

## **1.5 Organisation du manuscrit.**

Ce manuscrit est divisé en six chapitres. Le premier chapitre présente l'introduction du projet, en exposant le contexte, les motivations, les problématiques et les objectifs. Le deuxième chapitre aborde la structure d'accueil en décrivant l'entreprise hôte, ses missions, ses objectifs et les projets réalisés. Le troisième chapitre Expérience utilisateur tel que UX et UI, Le quatrième chapitre traite de l'environnement de travail en abordant des éléments tels que l'infrastructure matérielle, le système d'exploitation, la conception et la méthodologie de développement. Le cinquième chapitre se concentre sur la conception de l'application en détaillant la conception fonctionnelle et la conception de la base de données Le sixième chapitre les outils utilisés et technology tel que django, flutter, git et hitub Le septième chapitre décrit la réalisation de l'application en abordant le développement du backend, du frontend et le déploiement. Enfin, une conclusion générale et des références du projet.

# Chapitre2 : Structure d'accueille

## 2.1 Contexte

Dans ce chapitre dédié à la structure d'accueil, nous explorerons en détail l'entreprise Next-technology, qui a servi de cadre à notre stage. Nous présenterons les informations clés relatives à l'entreprise, telles que son domaine d'activité, ses missions et ses objectifs. Nous mettrons également en évidence les projets réalisés par Next technology, témoignant de son expertise et de son succès dans le secteur.

Ensuite, nous aborderons en détail le déroulement de notre stage au sein de l'entreprise. Nous décrirons les différentes étapes que nous avons traversées, les tâches qui nous ont été confiées et les responsabilités que nous avons assumées.

## 2.2 Présentation de l'entreprise

Next-Technology est une société mauritanienne spécialisée dans la digitalisation. Elle propose à ses clients des solutions logicielles de haute qualité pour répondre à leurs besoins. Leurs produits sont conçus en tenant compte d'une vision d'utilisation future ainsi que des tendances stratégiques du secteur financier. Leur objectif est de contribuer au positionnement de l'Afrique en tant qu'acteur clé du futur de la technologie financière. Depuis sa création sur le marché mauritanien, Next-Technology a réalisé plusieurs projets d'envergure pour des grands comptes en Mauritanie, grâce à une large gamme de services. Dans le cadre des projets qu'ils ont mis en œuvre.

1. DIGIPay : La première solution de banque numérique développée en Mauritanie a déjà été déployée par plusieurs institutions financières.
2. Next-Connect: Actuellement déployé chez Gimtel sous le nom de GimtelConnect, l'objectif est de simplifier le processus de paiement entre les banques et les facturiers.
3. FastPay : Plateforme d'intégration de solutions de paiement pour les commerçants dotée d'un système innovant pour le e-commerce. [BB]
4. DVS31 : Solution de cash flow management pour les entreprises spécialement conçue pour les réalités du marché africain.
5. Sedad Bank : Sedad Bank est la première application de wallet développée en Mauritanie et lancée par la Banque de Mauritanie. j'ai développé ce produit pour la BMI
6. Bimbank: est une application de wallet développée en Mauritanie et lancée par la Banque de Mauritanie. j'ai développé ce produit pour la BMI



### **2.3 Méthodologie du travail :**

Le choix de la méthode revêt une importance cruciale pour obtenir un développement de haute qualité. Le principe sous-tendant cette méthodologie de travail vise à réduire la complexité du système en proposant des procédures traçables et des étapes précises. Une méthode d'analyse ou de conception représente un processus destiné à formaliser la phase de développement en vue de fidéliser la clientèle. Nous entamons cette démarche par une déclaration informelle.

La phase d'analyse peut énumérer les résultats escomptés en termes de fonctionnalités, de robustesse, de sécurité, de maintenance, d'évolutivité, etc. Dans cette phase, il est possible de décrire le fonctionnement futur du système de manière quel que peu ambiguë (généralement à l'aide d'un langage de modélisation) afin de faciliter sa mise en œuvre.

### **2.3 Choix d'une méthodologie de travail**

Pour la mise en œuvre de ce projet, j'ai opté pour la méthode utilisée par NEXT Technologie, à savoir la méthodologie Agile Scrum. La méthodologie Agile Scrum est une approche de gestion de projets informatiques. Le terme "Scrum" est emprunté au rugby, faisant référence à une réunion informelle tenue quotidiennement pour discuter des réalisations et des plans pour le lendemain et au-delà.

Afin d'assurer une régularité et de réduire le besoin de réunions fréquentes, Scrum est structuré en événements. L'un de ces événements est le sprint, qui représente une itération dans le processus de développement d'un projet. En d'autres termes, un sprint est une courte période de travail planifié que l'équipe s'engage à accomplir et à soumettre pour examen.

Les sprints sont généralement de courte durée, souvent entre deux et quatre semaines. L'objectif du sprint est établi de manière collaborative entre le propriétaire du produit et l'équipe, et il est connu sous le nom d'"objectif du sprint". Le travail à réaliser provient du backlog de sprint, qui est une liste de tâches à accomplir.

La méthodologie Scrum repose sur trois rôles principaux :

- Le Product Owner, qui se focalise sur les aspects commerciaux du projet.
- Le Scrum Master, qui agit en tant que coach et facilitateur.
- L'équipe, qui travaille de manière autonome en suivant les méthodes qui lui conviennent le mieux pour atteindre les objectifs fixés par le Product Owner.

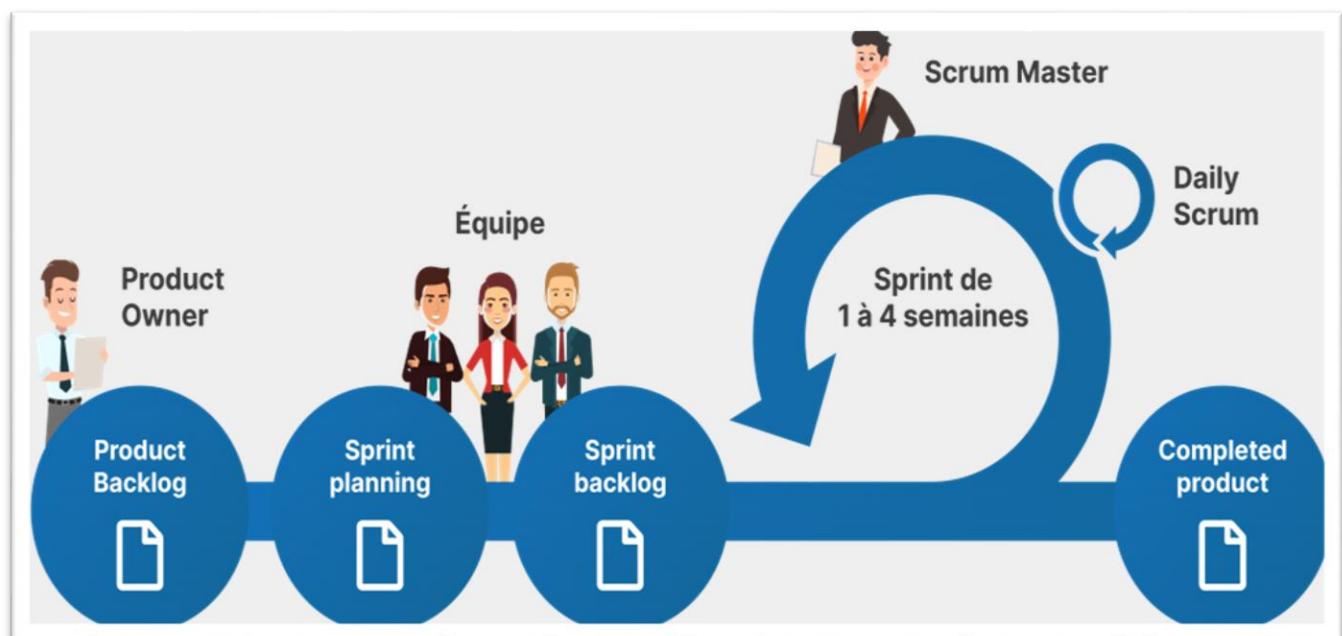


Figure 1 : le fonctionnement de la méthode scrum

Dans notre cas, lorsque j'ai voulu lancer ces sprints, j'ai utilisé un "Scrum Board"

Pour visualiser la progression de l'équipe.

Un Scrum Board (Scrum Task Board) est un outil qui aide les équipes à rendre visibles les éléments du Sprint Backlog. Le tableau peut prendre de nombreuses formes physiques (c'est-à-dire des tableaux blancs et des autocollants) et virtuelles (c'est-à-dire des outils mais peu importe son apparence, il remplit la même fonction). Le Scrum Board est le point central de tout projet agile et un endroit idéal pour organiser des réunions. Le tableau est mis à jour et référencé par l'équipe et affiche tous les projets lors de la mêlée quotidienne, il maintient l'équipe concentrée sur les tâches restantes et leurs priorités.

Next Technologie utilise le logiciel Trello pour la gestion de projet, avec lequel nous accomplissons les tâches dans les Scrum Boards que l'entreprise nous a confiés. Voici un exemple d'un tableau « Scrum Board » que j'ai adopté sur Trello:

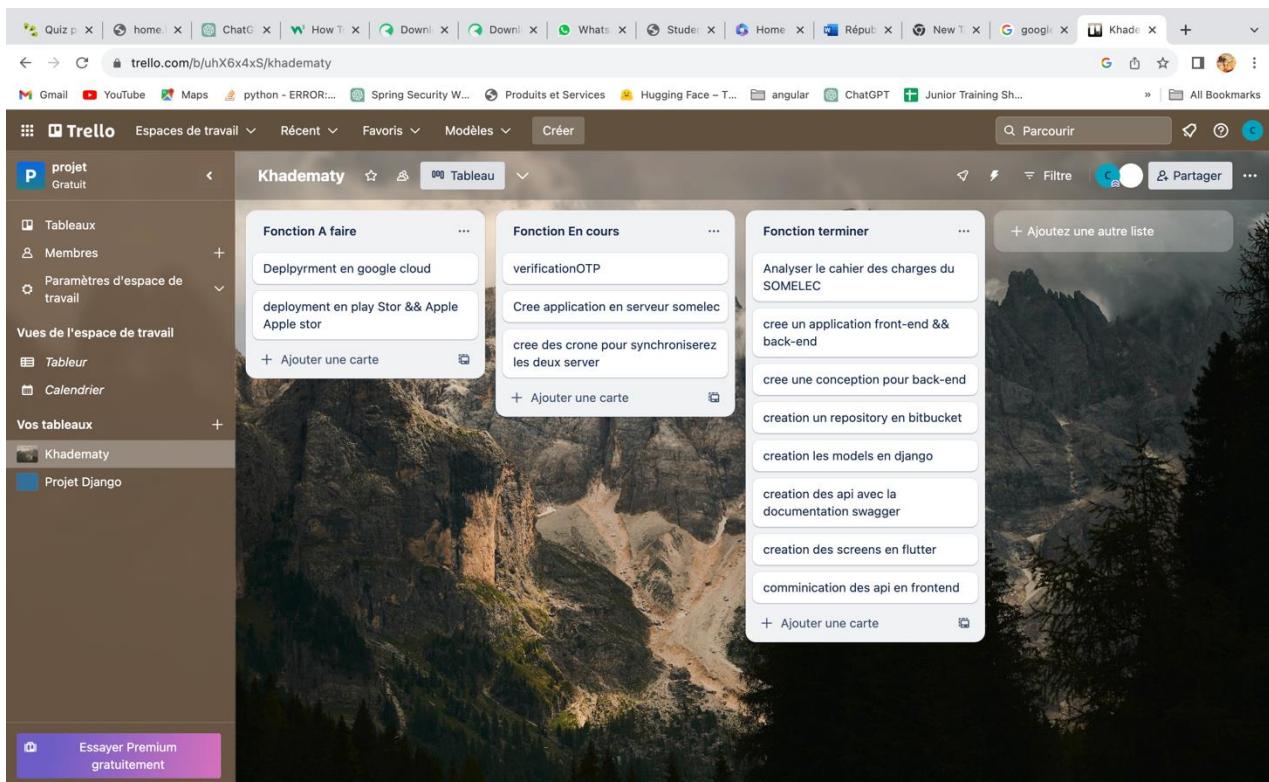


Figure 2: scrum board

Comme vous pouvez le voir sur la photo ci-dessus, les tâches à effectuer sont dans la colonne "Fonction Complet", et lorsque je lance une tâche, je déplace le post-it qui la représente dans la colonne "En cours". Lorsqu'ils terminent un élément du Sprint Backlog, les tâches vérifiées ont été déplacées vers " Fonction Complet" et j'ai sélectionné une nouvelle tâche. Chaque tableau Scrum dure pendant le Sprint. Un nouveau sprint nécessite que l'équipe crée un nouveau tableau Scrum ou réinitialise un tableau Scrum existant.

## **2.4 Expérience de stage**

Mon expérience de stage s'est déroulée dans un environnement très accueillant et professionnel. J'ai eu l'opportunité d'acquérir les compétences nécessaires à la réalisation de ce travail, ainsi que de développer de nombreuses connaissances qui me seront utiles au-delà de cette période de stage.

Le stage, quant à lui, peut être divisé en trois phases : la phase de formation pré-sujet, la phase de formation post-sujet et la phase de réalisation du sujet.

Examinons le déroulement de ces trois phases dans l'ordre chronologique :

- Phase de formation pré-sujet :

Avant de recevoir mon sujet de stage, il était nécessaire d'acquérir les bases nécessaires pour pouvoir le réaliser. Quel que soit le sujet, il sera réalisé en utilisant des outils et technologies disponibles dans Next-Technology. Ainsi, une période d'apprentissage et de familiarisation a débuté. J'ai découvert le fournisseur de cloud Heroku ainsi que plusieurs de ses modules les plus couramment utilisés, tels que Django, Flutter, Swagger, API Rest, et bien d'autres. Pour m'assurer de bien assimiler ces nouvelles connaissances, j'ai réalisé une mini application CRUD en utilisant toutes ces technologies. Cette application a servi à valider l'acquisition d'un niveau de base me permettant de passer à la phase suivante, c'est-à-dire l'obtention d'un sujet.

- Phase de formation post-sujet

Le sujet du stage a été choisi, qui est donc khadematy déjà évoqué.

Pour commencer le sujet du stage, une autre phase de formation s'imposait. En effet, étant donné que des technologies totalement nouvelles pour moi allaient être utilisées dans la réalisation de ce système, il était nécessaire de les apprendre. Le même schéma que dans la phase précédente a donc été suivi : apprentissage, suivi d'un mini-projet pour s'assurer d'atteindre un niveau minimal permettant de commencer le travail sur le sujet. Ensuite, une fois cet objectif atteint, nous sommes passés à la dernière phase du stage : la réalisation de l'application.

- Phase de réalisation du sujet :

Cette phase concerne la réalisation du sujet, en commençant par la conception et la modélisation, puis en passant au développement de toutes les fonctionnalités du backend et en les reliant à l'interface frontend. Pendant cette période

Enfin, une fois l'application développée, elle a été déployée pour être opérationnelle.

L'environnement de mon stage m'a offert donc un cadre propice à l'apprentissage et au développement personnel, où j'ai pu évoluer aux côtés de professionnels expérimentés. Cette expérience enrichissante m'a permis d'approfondir mes compétences techniques, d'améliorer ma capacité à travailler en équipe et de développer ma compréhension du domaine. Je suis reconnaissant d'avoir eu l'opportunité de faire ce stage dans un environnement aussi favorable à mon apprentissage et à ma croissance professionnelle.

# Chapitre 3 : Expérience utilisateur

## 3.1 Introduction

Le but de ce chapitre est d'expliquer tous les éléments liés à l'UX en décrivant les différentes étapes du processus UX suivies lors de la mise en place de ma solution. Enfin, je montrerai une capture de la conception ou du prototype que j'ai implémenté avant de commencer à coder.

Le processus vise à augmenter la satisfaction des utilisateurs en améliorant la convivialité, l'accessibilité et le plaisir d'interagir avec le produit. Il englobe de nombreuses dimensions et comprend un ensemble diversifié de disciplines.

L'expérience utilisateur vise à fournir un produit utile, utilisable et agréable, créant ainsi une relation positive et solide entre l'entreprise qui le fournit et les personnes qui l'utilisent.

## 3.2 Le processus de UX

L'expérience utilisateur utilise un processus de conception centrée sur l'utilisateur (UCD) pour atteindre ses objectifs. L'idée principale de ce processus est d'impliquer le client dès le début de la conception en utilisant une variété de techniques de recherche pour mieux comprendre le problème.

Les étapes de l'UCD sont les suivantes :

**1. Compréhension :** Les chercheurs utilisateurs définissent le problème en comprenant le domaine et les utilisateurs. Le chercheur utilisateur suit l'utilisateur, pose des questions pour trouver des réponses et adopte une approche de débutant. À partir de cela, il commence à observer les modèles de comportement de l'utilisateur et en tire des conclusions sur ce que l'utilisateur veut, dont il a besoin ou l'idée qu'il cherche. C'est ainsi qu'il définit le problème.

**Outils disponibles à ce stade :**

- **Persona :** Il décrit en détail qui sont les principaux clients et pour quoi ils se sentent ainsi.
- **Customer journey map (cartes de parcours client) :** Elle montre ce que font vos clients lorsqu'ils interagissent, y compris ce qui répond à leurs attentes, ce qui ne répond pas à leurs attentes et ce que l'organisation peut améliorer pour mieux servir les clients.

**2. Recherche (Research):** Cette étape vise à comprendre ce que les utilisateurs veulent dans une conception. Il existe de nombreuses façons d'aider à cela, notamment :

- Entretiens

- Groupes de discussion

- Sondages

**3. Esquisser (Sketch):** Les concepteurs d'interaction clés identifient des solutions pour résoudre ce problème et créent des esquisses conceptuelles. Cela commence par divers concepts dessinés à la main, qui sont ensuite transformés en wireframes plus précis.

**4. Concevoir (Design):** Une autre étape clé de l'ensemble du processus consiste à rendre la solution esthétique. C'est le travail du concepteur visuel qui crée l'apparence de l'application, en se concentrant sur des éléments tels que la mise en page, la palette de couleurs et les caractères du logo.

**5. Test:** Cette étape est en réalité présentée sous toutes les autres étapes et vise à mettre le produit devant les utilisateurs dès que possible et à l'améliorer en fonction de leurs retours.

C'est à ce niveau que j'ai réalisé des prototypes qui ont été validés par des personnes expérimentées.

Les figures ci-dessous présentent mon prototype.

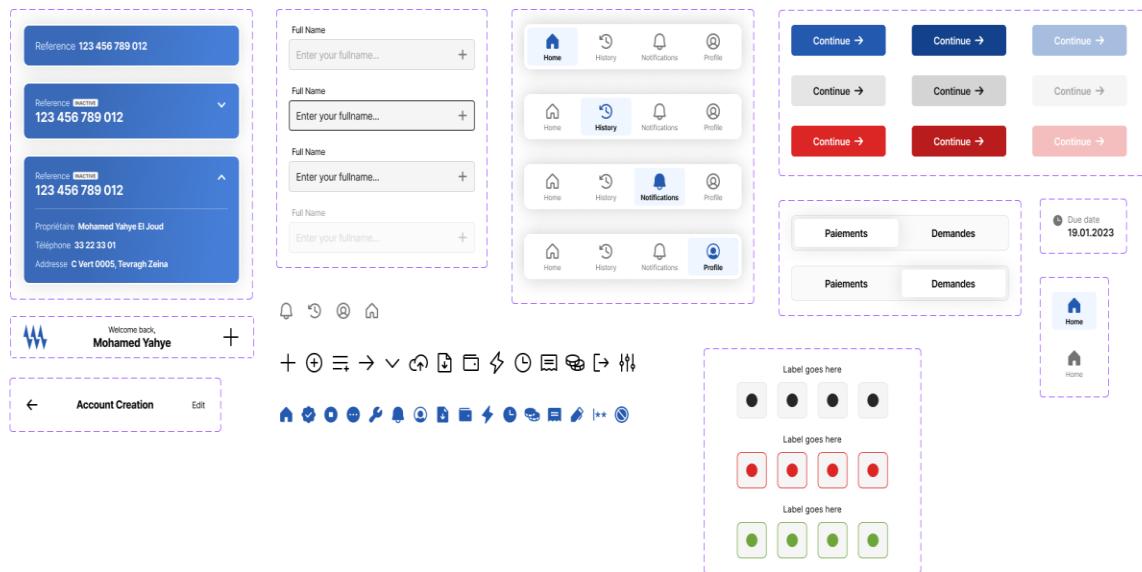


Figure 3 : composantes de mon design d'utilisation

## SIGNIN & SIGNUP

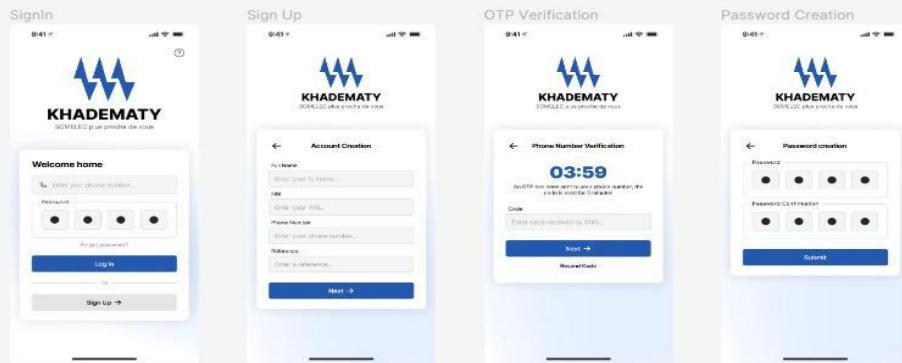


Figure 4: page d'authentification

## FORGOT PASSWORD

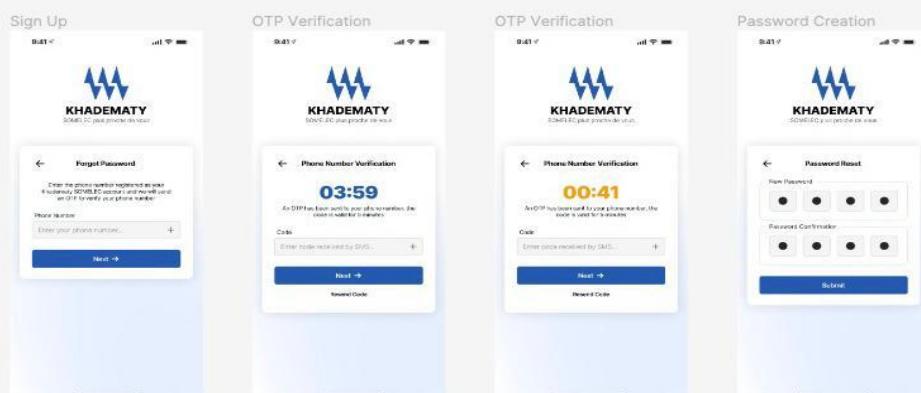


Figure 5: forget password

# HOME (NO REF.)

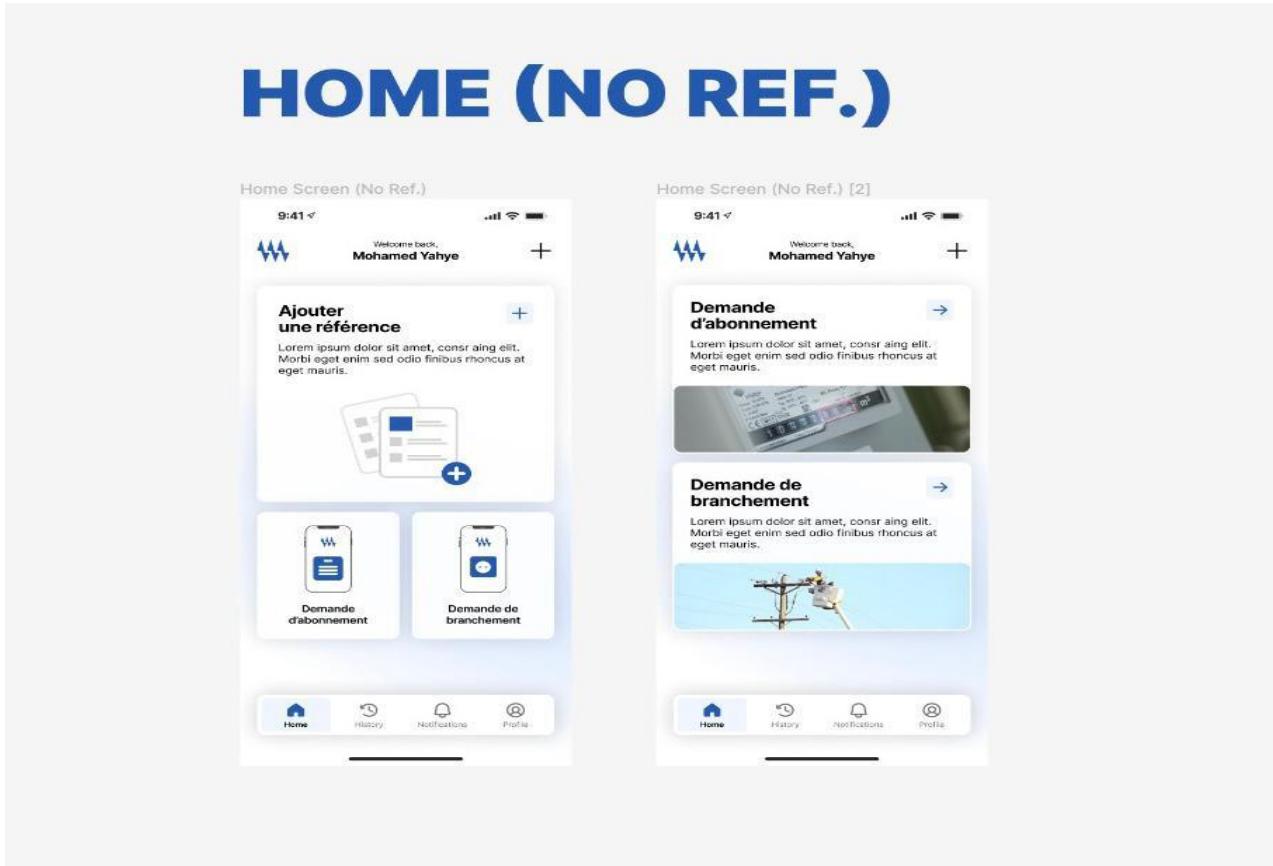


Figure 6: home (no ref)

# HOME (RÉF.)

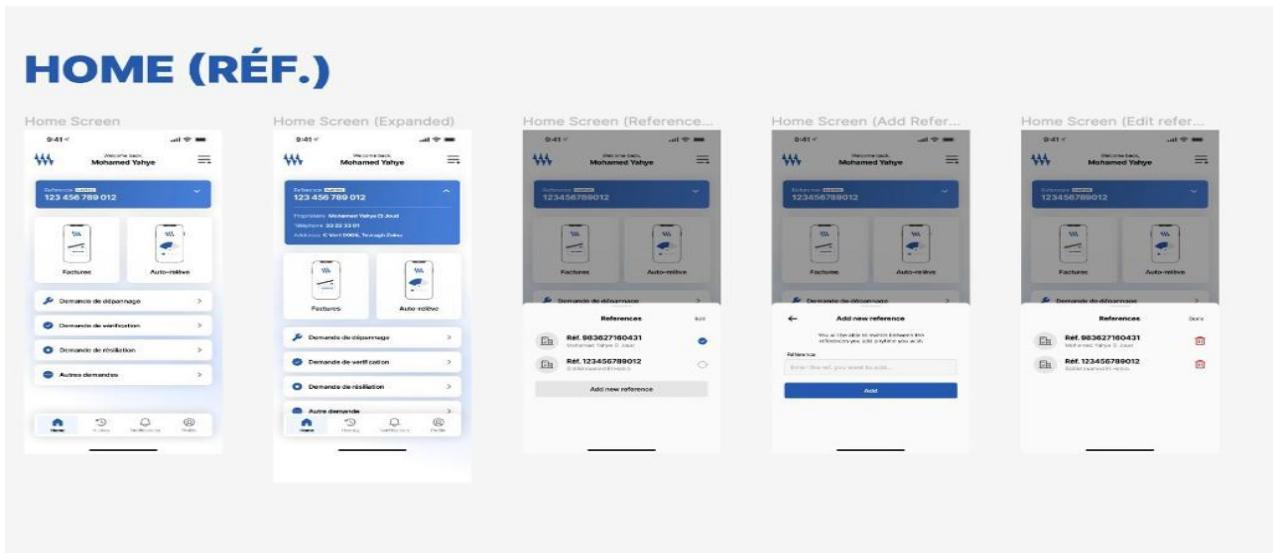


Figure 7: home(ref)

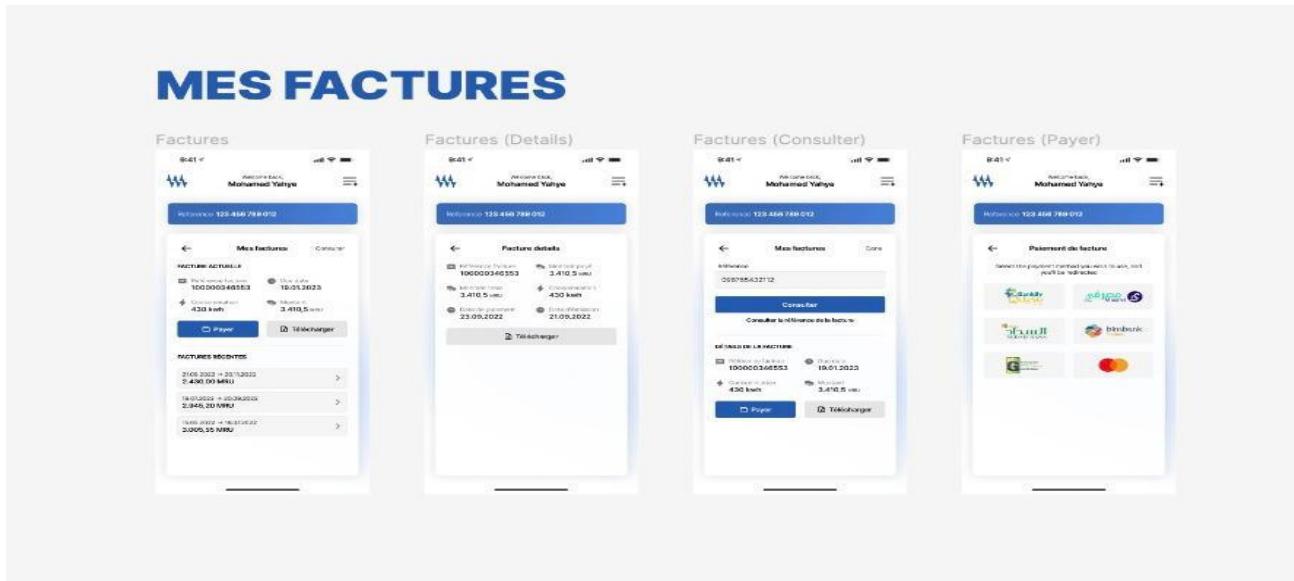


Figure 8: mes Facture

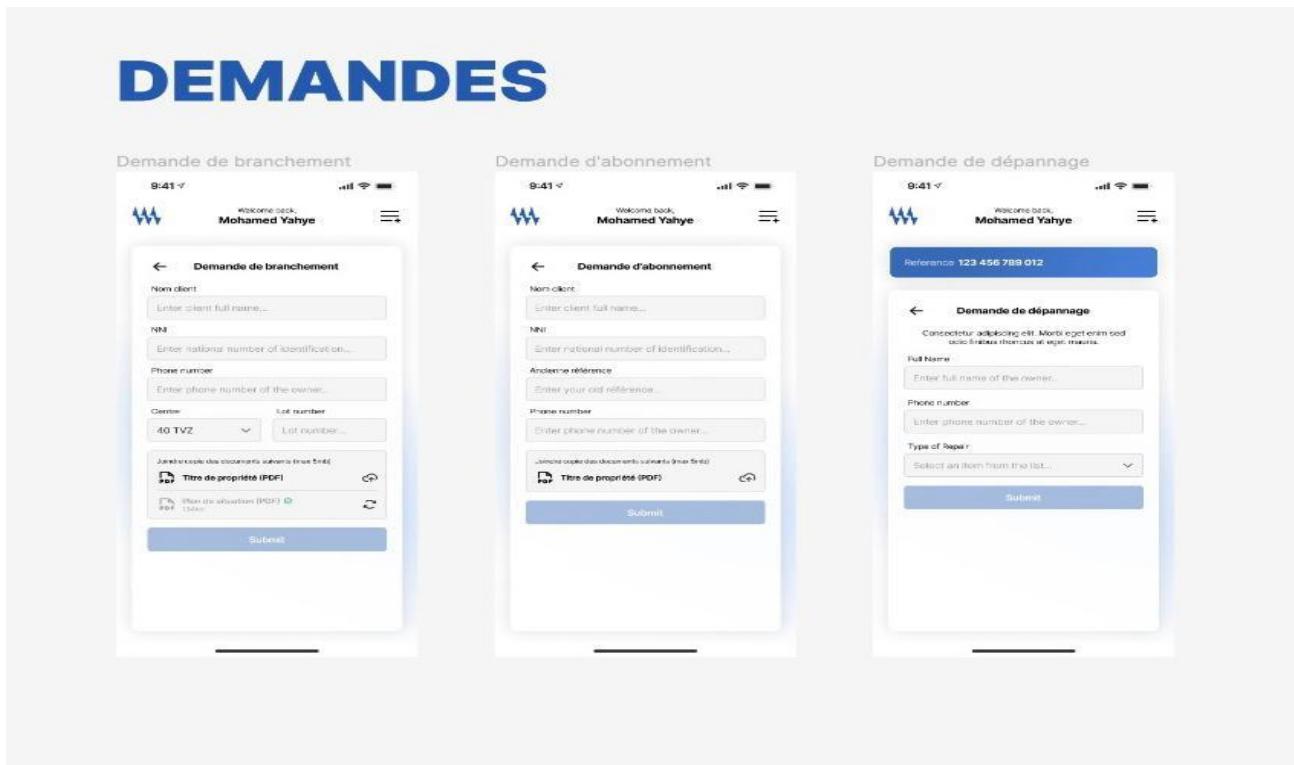


Figure 9: demandes

# AUTO-RELEVE

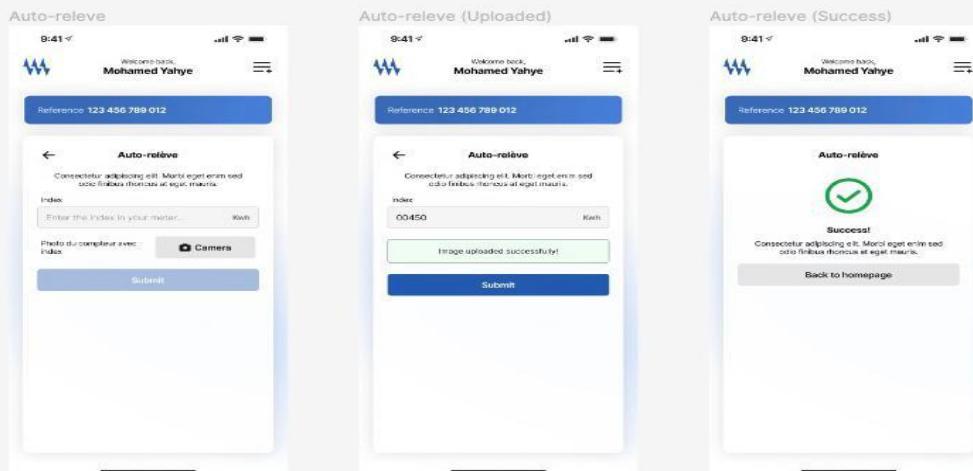


Figure 10: auto-releve

# HISTORY

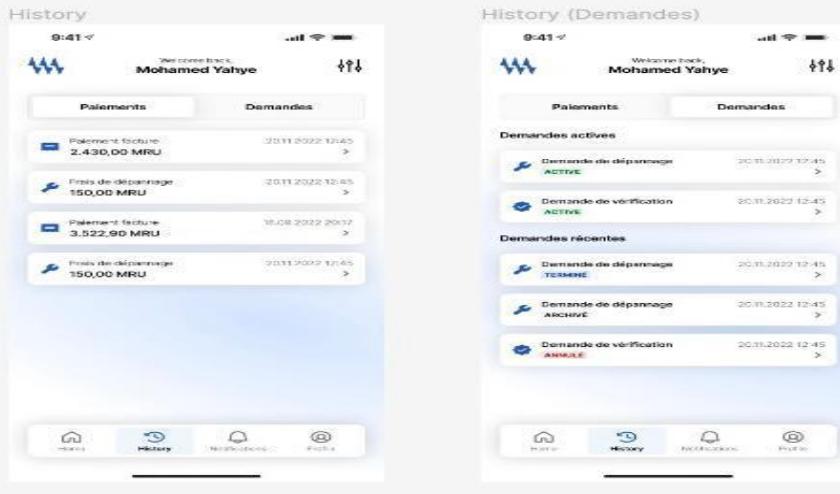


Figure 11: history

# PROFILE

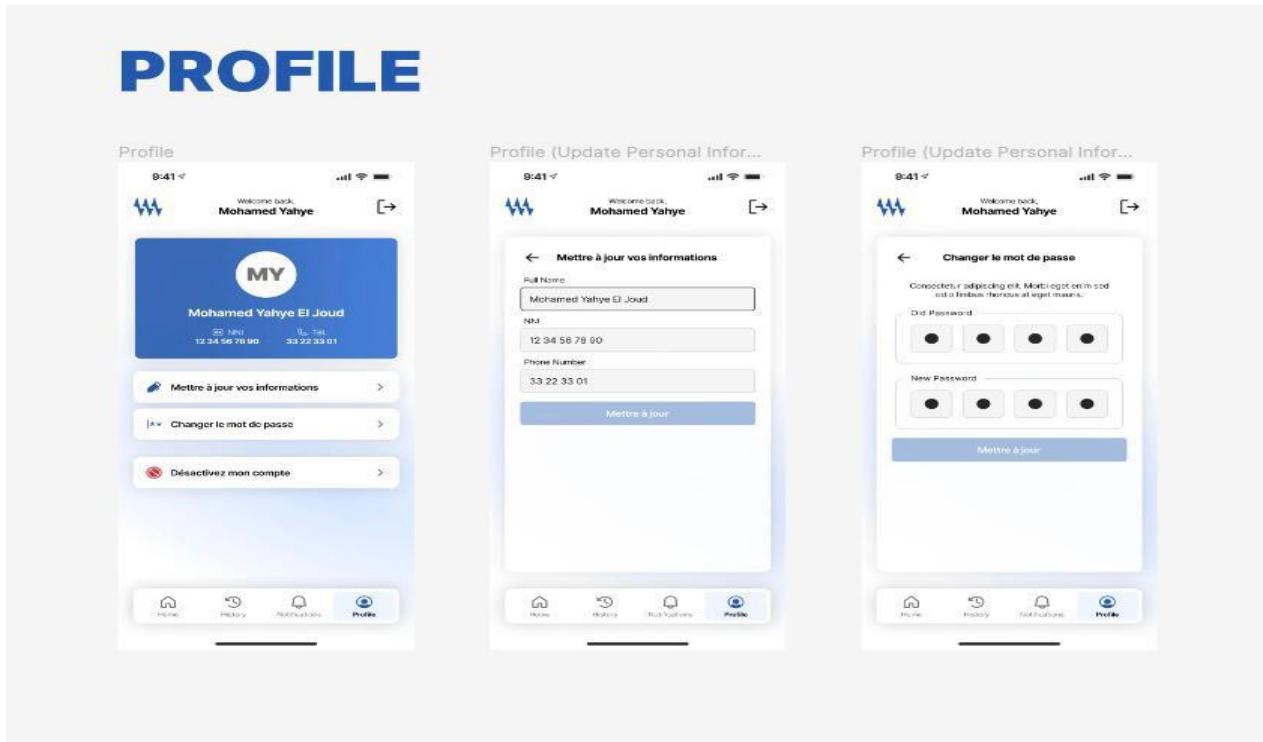


Figure 12 : profile

# Chapitre 4 : Environnement de travail

Ce chapitre se concentre sur l'environnement de travail et les outils utilisés tout au long de la réalisation de ce mémoire, couvrant ses différentes phases.

## 4.1 Infrastructure matérielle.

L'infrastructure matérielle de l'appareil utilisé lors du stage est composée des éléments suivants :

- Processeur : 3,5 GHz Dual-Core Intel Core i7
- Mémoire RAM installée : 16 GB 2133 MHz LPDDR3
- Type du système : macOS Ventura 13.0.1 (22A400) [OB]

## 4.2 Système d'exploitation

Le système d'exploitation de l'appareil utilisé lors du stage est MacOS Ventura Famille. Voici quelques détails supplémentaires :

- Version: 13.0.1 (22A400)
- Date d'installation : 2017, Four Thunderbolt 3 Ports
- Numéro de build: 499.40.2.0.0

## 4.3 Conception

Dans cette section, nous abordons les différents outils utilisés pour réaliser les différents types de modélisations de notre application. [OB]

### 4.3.1 Modelisation

Dans cette section, nous décrivons le langage de modélisation que nous avons utilisé pour concevoir l'application. Ensuite, nous présentons le logiciel que nous avons utilisé pour réaliser les différents diagrammes.

### 4.3.2 Langage de modélisation

Le langage de modélisation que nous avons utilisé est UML. Il fournit une modélisation orientée objet, à base de diagrammes, permettant une représentation schématique de l'application sous différentes perspectives. Nous avons utilisé le diagramme de cas d'utilisation. Ce diagramme aide à concevoir les exigences fonctionnelles de l'application.

### 4.3.3 Logiciel de modélisation

Le logiciel de modélisation que nous avons utilisé est StartUML. StartUML est une application en ligne qui permet de créer des diagrammes de manière intuitive et facile. Il offre une large

gamme de fonctionnalités pour la création de différents types de diagrammes, y compris les diagrammes UML, les diagrammes de flux, les diagrammes de réseau, les organigrammes, etc.

#### 4.3.4 Motif d'architecture

Dans cette section consacrée à l'architecture, nous allons présenter l'architecture MVC (Modèle-Vue-Contrôleur) pour le back end utilisée dans notre projet.

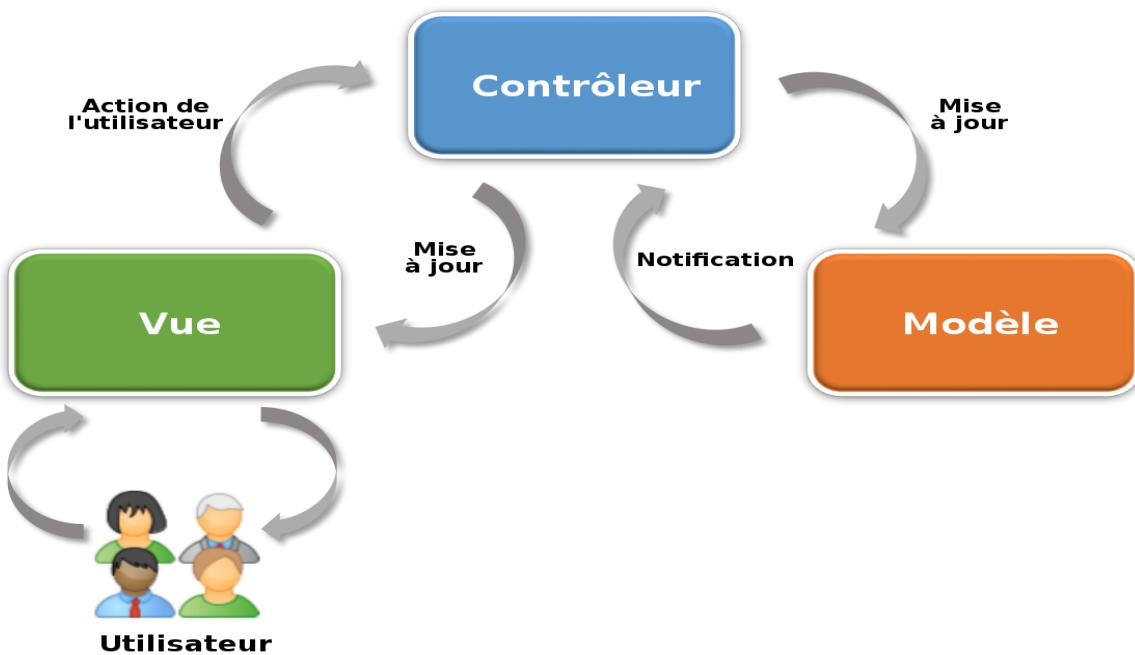
L'architecture MVC : est un modèle de conception largement utilisé dans le développement d'applications web. Elle divise une application en trois composants principaux :

- Modèle (Model) : Il représente les données et la logique métier de l'application. C'est dans cette partie que les opérations de manipulation des données sont effectuées, telles que la récupération, la modification et la suppression.
- Vue (View) : Elle est responsable de l'affichage des données au format souhaité pour les utilisateurs. La vue est généralement une interface utilisateur conviviale avec laquelle les utilisateurs interagissent.
- Contrôleur (Controller) : Il agit comme un médiateur entre le modèle et la vue. Il reçoit les entrées de l'utilisateur, effectue les actions correspondantes sur le modèle et met à jour la vue en conséquence.

L'architecture MVC favorise la séparation des préoccupations et permet une meilleure organisation du code. Elle facilite également la maintenance et l'évolutivité de l'application.

Dans notre projet, nous avons utilisé l'architecture MVC pour structurer le flux de données et les interactions entre les différentes parties de l'application. Cette approche nous a permis de séparer clairement les responsabilités, d'améliorer la maintenabilité du code.

La figure 1 qui suit en présente une illustration.



*Figure 13 : Architecture MVC*

## 4.6 Développement

Cette section met l'accent sur les outils et les technologies que nous avons utilisés pour créer notre application. Nous avons soigneusement sélectionné ces outils pour garantir un développement efficace, une performance optimale et une expérience utilisateur de qualité.

### 4.6.1 Backend

Dans cette sous-section, nous allons aborder les différents outils et technologies que nous avons utilisés pour le développement de notre backend.

#### 4.6.1.1 Framework de backend

Dans le développement de notre backend, nous avons utilisé le framework Django, qui est basé sur Python, un langage de programmation minimaliste et flexible. Cela nous a permis de construire facilement des applications robustes et performantes.

Avec Django, nous avons pu gérer les routes, les middlewares et les requêtes HTTP de manière simple et efficace. Il offre une grande flexibilité en termes de configuration et de structure de projet, ce qui nous a permis de personnaliser notre application en fonction de nos besoins spécifiques.

#### 4.6.1.2 Système de gestion de base de données

Pour notre système de gestion de base de données (SGBD), nous avons utilisé PostgreSQL. Il est un système de gestion de base de données relationnelle open source très populaire. Il est souvent abrégé en "Postgres". PostgreSQL est réputé pour sa fiabilité, sa sécurité et sa puissance, et il est largement utilisé dans de nombreuses applications, allant des petites applications web aux grandes entreprises.

Voici quelques caractéristiques clés de PostgreSQL :

- **Architecture Avancée** : PostgreSQL utilise une architecture multi-versionnelle (MVCC) qui permet à plusieurs versions de données d'exister simultanément dans la base de données. Cela offre une meilleure gestion des transactions et des performances accrues dans les environnements concurrents.
- **Extensibilité** : PostgreSQL prend en charge les extensions personnalisées qui permettent aux utilisateurs de créer et d'ajouter de nouvelles fonctionnalités à la base de données. Cela signifie que vous pouvez personnaliser PostgreSQL en fonction des besoins spécifiques de votre application.
- **Langage de Programmation** : PostgreSQL prend en charge plusieurs langages de programmation pour écrire des fonctions stockées et des procédures, notamment PL/pgSQL, PL/Python, PL/Perl, PL/Tcl, etc.
- **Types de Données Étendus** : En plus des types de données standards, PostgreSQL offre une variété de types de données étendus, tels que les tableaux, les types géospatiaux (PostGIS) et les types de données personnalisés.
- **Sécurité** : PostgreSQL offre des fonctionnalités de sécurité avancées, notamment l'authentification personnalisable, le cryptage des données et la gestion fine des priviléges.
- **Support JSON** : PostgreSQL prend en charge le stockage et la recherche de données au format JSON. Cela est utile pour les applications qui traitent des données semi-structurées.
- **RéPLICATION et Haute Disponibilité** : PostgreSQL prend en charge la réPLICATION et la haute disponibilité via des solutions telles que la réPLICATION de flux, la réPLICATION logique et la mise en cluster.
- **Performances** : PostgreSQL offre d'excellentes performances pour les charges de travail lourdes et complexes grâce à ses optimisations internes et à sa prise en charge du parallélisme.

- **Communauté Active** : PostgreSQL dispose d'une communauté active de développeurs et d'utilisateurs qui contribuent constamment à son amélioration et fournissent un support et des ressources en ligne.

## 4.6.2

## Front-end

Pour le développement frontend de notre application, nous avons utilisé Flutter, un framework basé sur le langage de programmation Dart, populaire pour la construction d'interfaces utilisateur interactives. Avec son modèle de développement, nous avons pu créer une interface réactive et modulaire. Grâce à sa structure de composants réutilisables, nous avons facilité la maintenance et l'extension de notre code. [\[OB\]](#)

### 4.6.2.1 Motif d'architecture Front-end

Dans cette section consacrées à l'architecture, nous allons présenter l'architecture cubit, L'architecture de gestion d'état Cubit est un modèle de gestion d'état en Flutter qui offre une approche organisée et prévisible pour gérer les changements d'état dans une application. Cubit a été introduit par la bibliothèque Flutter\_bloc et est conçu pour aider à séparer la logique métier de l'interface utilisateur, tout en fournissant une manière cohérente et efficace de gérer l'état de l'application.

Voici comment fonctionne l'architecture de gestion d'état Cubit en Flutter :

- **Cubit**: Un Cubit est une classe qui gère un certain état et expose des méthodes pour mettre à jour cet état. Il agit comme le cerveau de la logique métier de l'application. Un Cubit peut émettre un nouvel état en réponse à une action de l'utilisateur ou à des modifications internes. Par exemple, si vous avez une application de liste de tâches, vous pouvez avoir un Cubit qui gère l'état de la liste de tâches.
- **State** : L'état est l'ensemble des données qui définissent l'état actuel de l'application. Dans le cas de Cubit, l'état est généralement représenté par une classe qui contient les différentes propriétés nécessaires pour décrire l'état de l'application à un moment donné.
- **Event** : Un événement est une action qui déclenche un changement d'état dans le Cubit. Par exemple, dans une application de liste de tâches, l'utilisateur peut cliquer sur un bouton pour ajouter une nouvelle tâche. Cela déclencherait un événement "Ajouter une tâche" qui serait traité par le Cubit pour mettre à jour l'état en conséquence.
- **Mapper** : Les mappers sont des fonctions utilisées pour transformer un événement en un état. Ils prennent l'état actuel et l'événement en entrée, puis retournent le nouvel état en fonction de l'événement. Les mappers permettent de séparer la logique de traitement des événements de la logique de gestion des états.
- **Blocs** : Les blocs agissent comme des gestionnaires d'événements et d'états. Ils écoutent les événements émis par l'interface utilisateur ou d'autres sources et utilisent les Cubits appropriés pour gérer les changements d'état en conséquence. Les

blocs aident à connecter les interactions de l'utilisateur avec la logique métier et à gérer les flux de données entre l'interface utilisateur et les Cubits.

L'architecture de gestion d'état Cubit en Flutter favorise la séparation des préoccupations et facilite la maintenance et le test de l'application. Elle permet également d'adopter une approche réactive pour gérer les changements d'état, ce qui peut améliorer les performances et l'expérience utilisateur. Ce pendant, il est important de comprendre les concepts de base de Cubit et de blocs pour tirer le meilleur parti de cette architecture dans le développement de vos applications Flutter.

La figure 2 qui suit en présente une illustration.

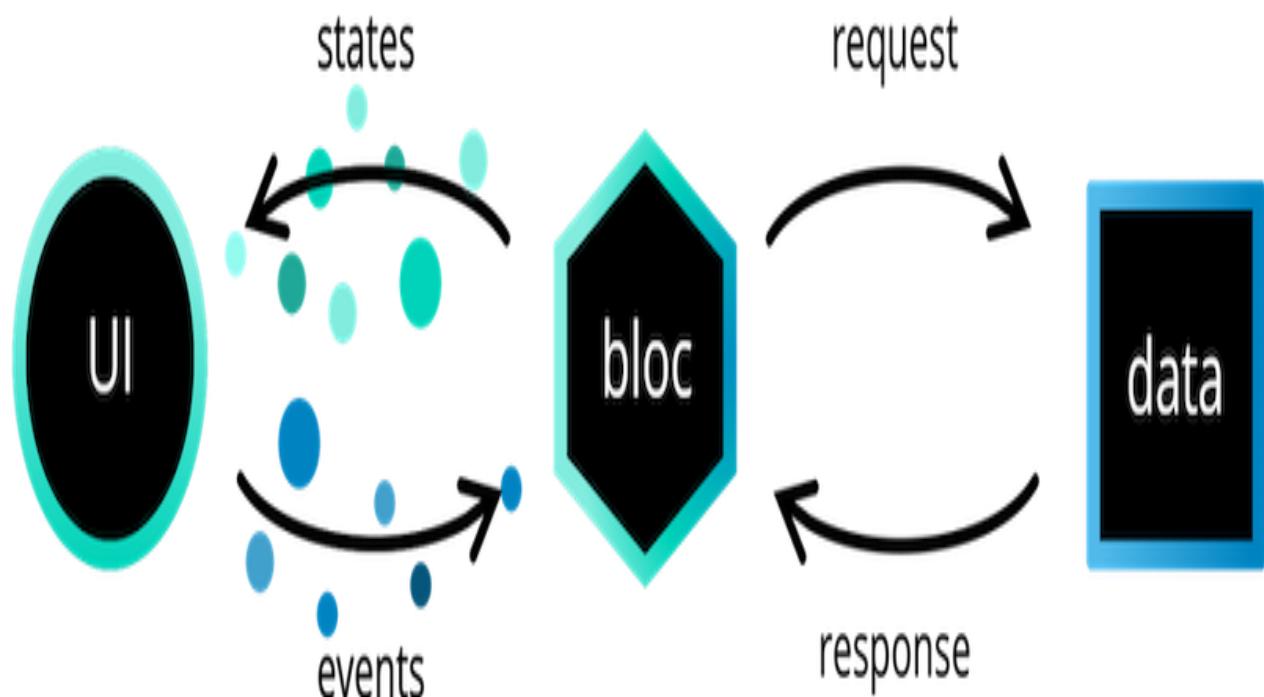


Figure 14: Architecture Cubit

#### 4.6.3 Ide



Dans notre projet, nous avons utilisé Visual Studio Code comme environnement de développement intégré (IDE). Visual Studio Code est un-IDE populaire et largement utilisé dans l'industrie du développement logiciel. Il offre de nombreuses fonctionnalités avancées pour faciliter l'écriture, la modification et le débogage du code.

Visual Studio Code prend en charge la syntaxe de toutes les technologies que nous avons utilisées dans notre projet, ce qui facilite grandement le développement.

Une des forces de Visual Studio Code réside dans sa richesse en plugins et extensions. Nous avons utilisé différents packages et plugins pour améliorer notre flux de travail et personnaliser l'expérience de développement selon nos besoins.

#### **4.6.4 Gestionnaires de dépendances**

PIP, acronyme de "Pip Installs Packages", est le gestionnaire de paquets et de dépendances standard pour Python. Il est largement utilisé pour installer, mettre à jour et gérer les packages Python et leurs dépendances. PIP facilite la distribution, l'installation et la gestion des bibliothèques et des modules tiers créés par la communauté Python et d'autres développeurs.

Lorsque vous installez un package à l'aide de PIP, il peut avoir des dépendances sur d'autres packages. PIP gère automatiquement ces dépendances et les installe en même temps que le package principal. Cela facilite la gestion des dépendances et assure que toutes les bibliothèques requises sont disponibles.

PIP est souvent utilisé en conjonction avec les environnements virtuels (virtual environments) pour isoler les packages et les dépendances d'un projet particulier. Les environnements virtuels permettent d'éviter les conflits entre les versions de packages et d'assurer que chaque projet a son propre espace isolé pour les dépendances.

Et à la fin on a créer un fichier "requirements.txt" pour lister toutes les dépendances de votre projet. Ce fichier peut ensuite être partagé avec d'autres développeurs pour garantir qu'ils installent les mêmes versions de packages. Vous pouvez générer ce fichier à partir de votre environnement en utilisant la commande **pip freeze > requirements.txt**.

Et lorsque le projet doit être installé dans un nouvel environnement, il suffit de lancer la commande `pip install -r requirements.txt` pour que le projet devienne capable d'être exécuté dans le nouvel environnement.

#### **4.6.5 Gestionnaire des versions**

Dans notre projet, nous avons utilisé Git comme gestionnaire de version. BitBucket est une plateforme populaire qui facilite le contrôle de version et la collaboration sur des projets de développements logiciel.

En utilisant bitbucket, nous avons créé un référentiel (repository) pour notre projet, où nous avons stocké et organisé notre code source. Nous avons utilisé des branches pour travailler sur des fonctionnalités ou des correctifs spécifiques sans affecter la

branche principale (généralement appelée "master" ou "main" ou n'importe quelle branche). Cela nous a permis de développer de manière itérative et d'intégrer les changements en toute sécurité dans le code principal lorsque les fonctionnalités étaient prêtes.

## 4.7 Déploiement

Dans le cadre du déploiement de notre application, nous avons utilisé une première phase, Heroku pour test, et une deuxième phase Google Cloud (GCP) et le serveur de Somelec.

Heroku est une plateforme de cloud computing qui permet aux développeurs de déployer, gérer et héberger des applications web de manière facile et rapide. Fondée en 2007, Heroku a gagné en popularité en tant que solution d'hébergement pour les applications web, en particulier les applications développées en langages tels que Ruby, Python, Node.js, Java, PHP, et bien d'autres.

Et nous avons utilisé Heroku lors de la phase de test de l'application.

Google Cloud Platform (GCP) est une suite complète de services de cloud computing proposée par Google. Elle permet aux entreprises et aux développeurs de déployer, gérer et échelonner des applications et des services en utilisant l'infrastructure et les services de Google. Voici quelques points clés sur Google Cloud Platform:

- **Infrastructure mondiale** : Google Cloud dispose d'un réseau mondial étendu de centres de données répartis dans différentes régions. Cela permet aux utilisateurs de déployer leurs applications et services dans des emplacements géographiques stratégiques pour une performance optimale et une faible latence.
- **Elasticité et mise à l'échelle** : GCP offre la possibilité de mettre à l'échelle automatiquement les ressources en fonction de la demande, ce qui permet aux utilisateurs de payer uniquement pour ce qu'ils consomment. Cela peut aider à gérer efficacement les pics de trafic et à réduire les coûts inutiles.
- **Big Data et Analyse** : GCP propose des outils puissants pour le traitement et l'analyse de grandes quantités de données, tels que BigQuery pour l'analyse de données en temps réel et Google Dataflow pour le traitement des flux de données.
- **Intelligence Artificielle et Machine Learning** : GCP offre des services d'intelligence artificielle et de machine learning, tels que TensorFlow et AutoML, permettant aux utilisateurs de créer des modèles d'IA et de ML personnalisés.
- **Sécurité et Conformité** : GCP accorde une grande importance à la sécurité et propose des fonctionnalités de sécurité avancées, y compris le chiffrement des données, l'authentification à deux facteurs et la conformité aux normes de l'industrie.
- **Outils de Développement** : GCP fournit des outils de développement tels que Google Cloud SDK pour gérer et déployer des applications, ainsi que des intégrations avec des outils populaires tels que Kubernetes et Docker.

- **Tarification flexible** : GCP propose une tarification flexible, y compris des options de paiement à l'utilisation et des modèles de tarification compétitifs pour répondre aux besoins et au budget des entreprises.
- **Services gérés** : GCP propose des services gérés tels que Google Kubernetes Engine (GKE) pour la gestion des conteneurs et Google App Engine pour le déploiement d'applications sans se soucier de l'infrastructure sous-jacente.

Mais Somelec souhaite que ses données soient confidentielles. Pour cela, nous avons créé une application Django au niveau du serveur. De plus, nous avons ajouté des tâches planifiées (cron jobs) pour assurer la synchronisation régulière. Nous avons également configuré le serveur pour n'accepter que les envois provenant de notre application.

## 4.8 Rédaction

Dans cette section, nous allons parler des outils utilisés pour la rédaction, que ce soit pour la création de ce rapport ou pour sa présentation.

### 4.8.1 Rapport

Pour la création du rapport, nous avons utilisé Microsoft Office Word. Word est un logiciel de traitement de texte développé par Microsoft, faisant partie de la suite Microsoft Office. Il offre de nombreuses fonctionnalités pour créer, éditer et formater des documents texte de manière professionnelle. Word propose des outils avancés tels que des modèles prédéfinis, des styles de mise en forme, des fonctionnalités de vérification orthographique et grammaticale, et bien plus. Grâce à ces fonctionnalités, nous avons pu concevoir et personnaliser le rapport en fonction de nos besoins spécifiques.

### 4.8.2 Présentation

Pour la présentation, nous avons utilisé Microsoft PowerPoint. Microsoft PowerPoint est un logiciel couramment utilisé pour créer des diapositives et des présentations visuelles. Il offre une large gamme de fonctionnalités pour créer des diapositives attrayantes, ajouter du texte, des images, des graphiques, des animations et des transitions.

# Chapitre 5 : Conception de l'application

Dans ce chapitre de conception, nous allons aborder les différentes étapes et considérations liées à la conception de notre projet

## 5.1 Rappel sur le besoin

Pour implémenter la conception, je vais utiliser le diagramme UML suivant :

- **Le Diagramme de cas d'utilisation** : utilisé pour donner une idée générale du fonctionnement du système. Il est utile pour les présentations aux parties prenantes dans les projets de développement.
- **Le Diagramme de séquence** : Il permet de montrer l'interaction des objets dans le cadre de la scène du diagramme de cas d'utilisation. Le but est de décrire comment des actions se produisent entre des acteurs ou des objets.
- **Le Diagramme de classes** : sont parmi les types les plus utiles de diagrammeUML, parce qu'ils indiquent clairement la structure d'un système, spécifique en modélisant les relations entre ses classes, propriétés, opérations et objets.
- **5.2 Diagramme de cas d'utilisation**
- Un Diagramme de cas d'utilisation permet d'obtenir une vue d'ensemble, de ce que le système ou une partie du système fait sans entrer dans les détails de la mise en œuvre. Notre système est assez grand, ce qui a rendu plus correct de diviser les diagrammes de Diagramme de cas d'utilisations en fonctions des gestions et fonctionnalités les plus différentes.

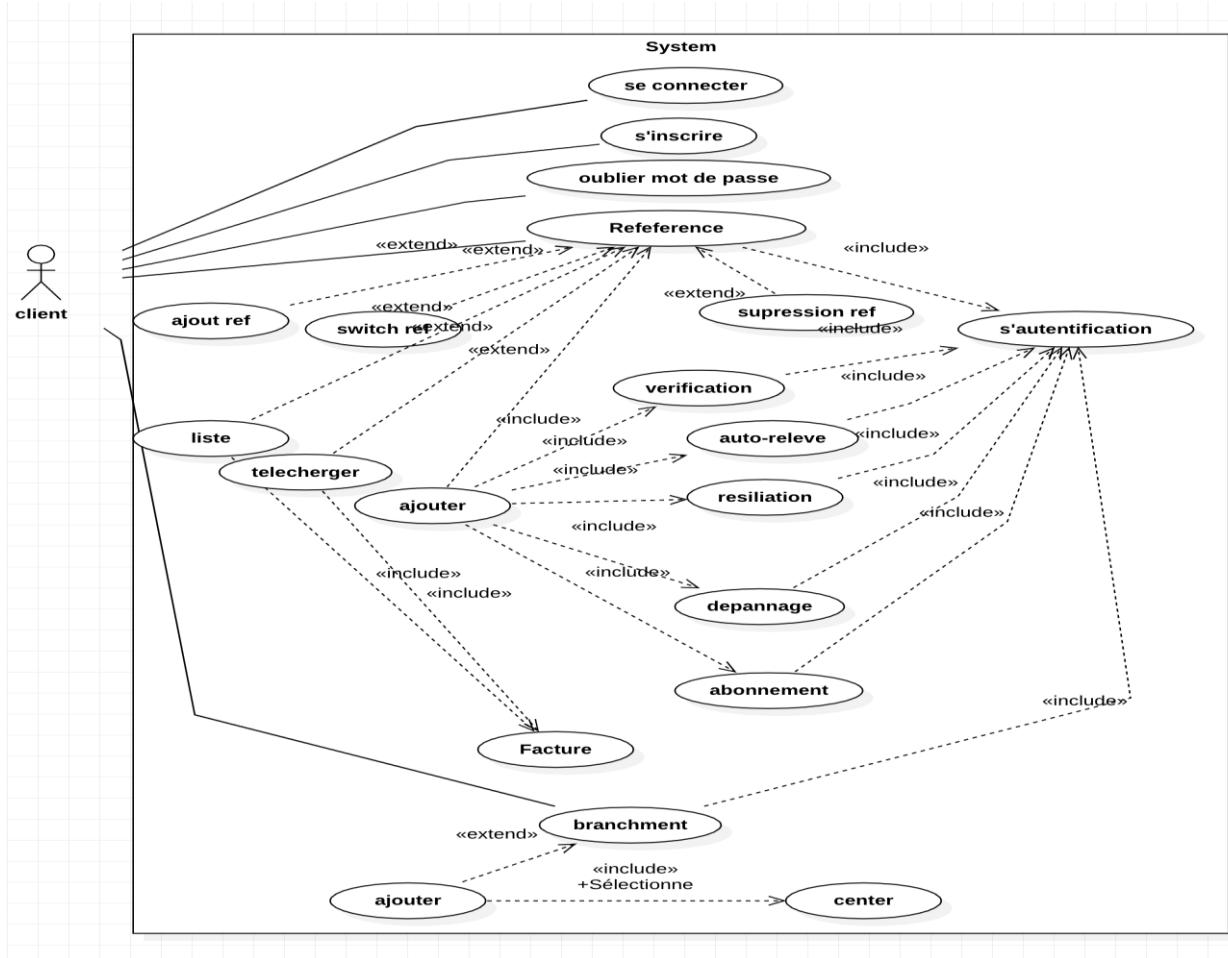


Figure 15 : diagramme de cas d'utilisation client

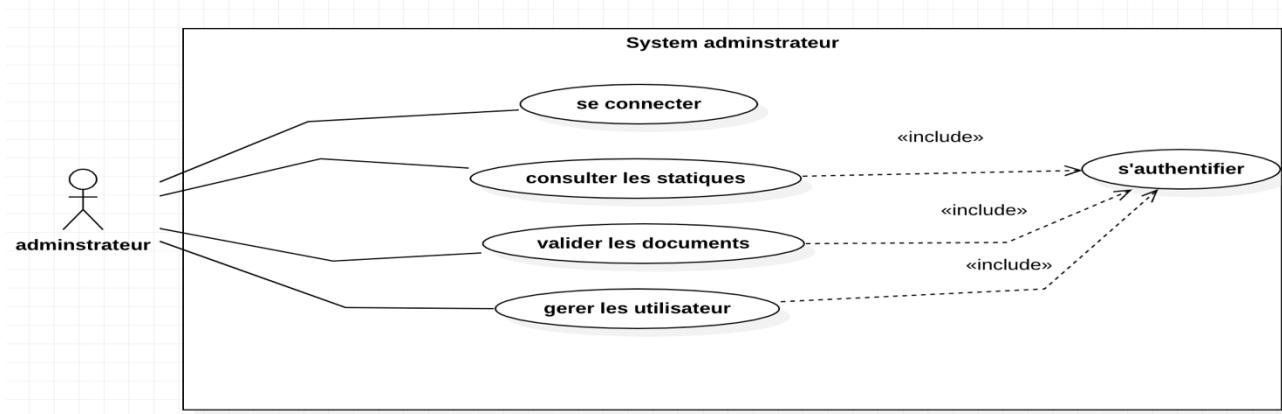


Figure 16 : diagramme de cas d'utilisation administrateur

#### 4.4 Diagramme de classe

Un diagramme de classe est un type de diagramme de modélisation en ingénierie logicielle qui permet de représenter les entités principales d'un système informatique, ainsi que leurs relations et leurs attributs. Contrairement au diagramme de cas d'utilisation, qui se concentre sur les interactions entre acteurs et cas d'utilisation, le diagramme de classe se penche sur la structure statique du système.

Plus précisément, un diagramme de classe identifie et décrit les classes (ou objets) qui composent le système, leurs propriétés (attributs) et leurs relations. Les principales composantes d'un diagramme de classe incluent :

**Classes** : Les entités principales du système sont représentées sous forme de classes. Chaque classe incarne un ensemble d'objets similaires ayant des caractéristiques communes.

**Attributs** : Les attributs sont les propriétés ou les données associées à chaque classe. Ils décrivent les caractéristiques spécifiques des objets de cette classe.

**Méthodes**: Les méthodes sont les opérations ou les fonctions que les objets de la classe peuvent effectuer. Elles définissent le comportement des objets.

**Relations** : Les relations entre les classes illustrent comment les différentes classes sont connectées ou interagissent les unes avec les autres. Les types de relations courantes incluent l'agrégation, la composition, l'héritage, l'association, etc.      **Multiplicité**: La multiplicité précise combien d'objets d'une classe peuvent être liés à un ou plusieurs objets d'une autre classe.  
**Généralisation/Héritage**: Cela indique comment les classes peuvent être organisées dans une hiérarchie, où les sous-classes héritent des propriétés et des méthodes de leurs classes parentes.

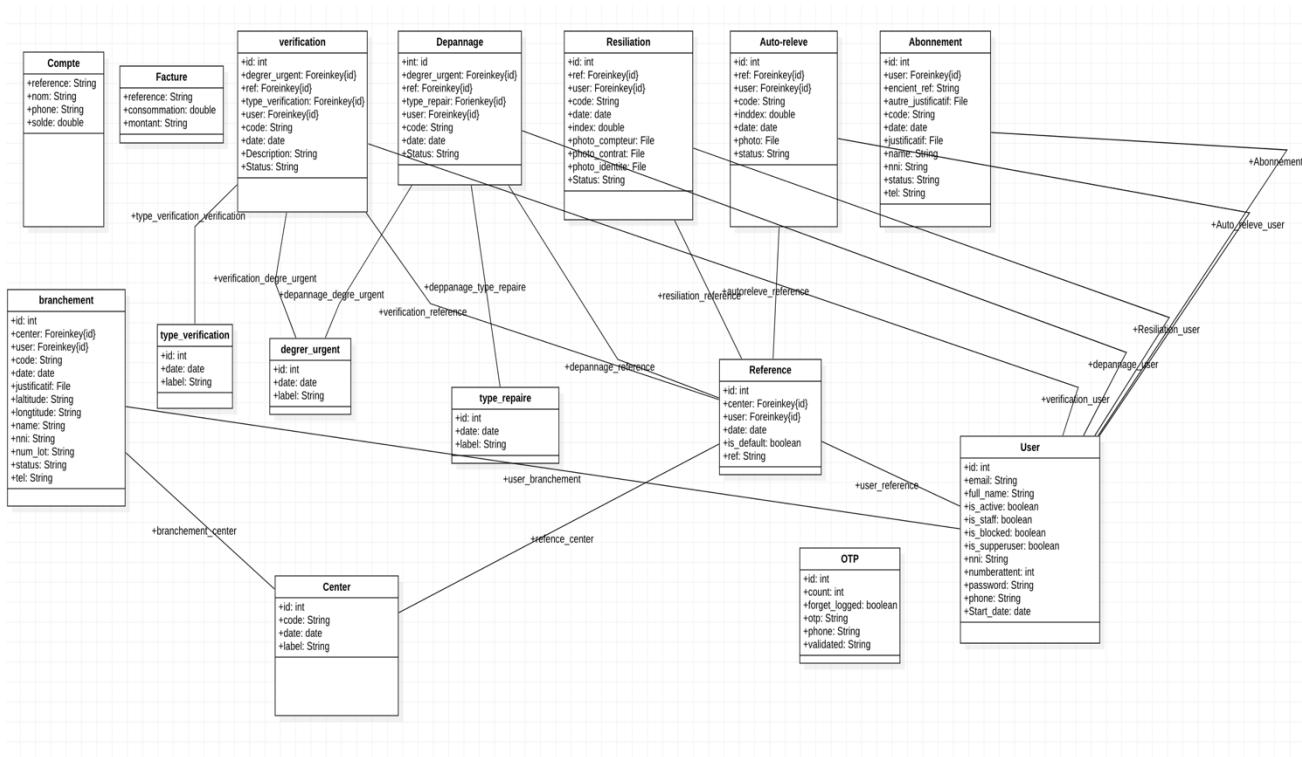


Figure 17 : diagramme de classe

# Chapitre 6 : Les soutils utilisés et technologies

## 6.1 Python



Python est un langage de programmation de haut niveau, polyvalent et très populaire. Le langage de programmation Python est utilisé dans le développement Web, les applications d'apprentissage automatique, les technologies dans l'industrie du logiciel

## 6.2 Django



Django REST Framework est un framework de développement web en Python. Il facilite la création d'applications web robustes en fournissant une structure organisée et des fonctionnalités intégrées pour gérer les tâches courantes de développement web.

## 6.3 Dart



Dart est un langage de programmation développé par Google. Il est principalement utilisé pour le développement d'applications mobiles et web, notamment avec le framework Flutter.

## 6.4 Flutter



Flutter est un framework de développement d'applications multiplateformes créé par Google. Il permet de créer des applications mobiles avec une seule base de code pour Android et iOS.

## 6.5 vs code



Visual Studio Code (VS Code) est un éditeur de code source léger et puissant développé par Microsoft. Il est populaire pour sa flexibilité et son support pour divers langages de programmation.

## 6.6 Android studio



Android Studio est l'environnement de développement officiel pour les applications Android. Il offre des outils complets pour concevoir, coder, tester et déployer des applications Android.

## 6.7 Postman



Postman est un outil de test et de développement d'API. Il permet de tester, de documenter et de partager les API de manière efficace.

## 6.8 Figma



Figma est un outil de conception d'interface utilisateur basé sur le web. Il permet aux designers de créer des maquettes, des prototypes et des interfaces collaboratives.

## 6.9 Start UML



Start UML est un outil de dessin en ligne pour créer des diagrammes et des schémas, notamment des diagrammes de flux, des organigrammes et des plans architecturaux.

## 6.10 Git



Git est un système de contrôle de version distribué utilisé pour suivre les modifications du code source et collaborer efficacement dans le développement logiciel.

## 6.11 Bitbucket



Bitbucket est une plateforme de gestion de code source et de collaboration, principalement utilisée pour l'hébergement de projets Git.

# Chapitre 7 : Le Réalisation de l'application

Dans ce chapitre, nous allons voir le travail réalisé lors de l'implémentation de l'application, en commençant par la partie backend jusqu'au déploiement, en passant par la partie frontend.

## 7.1 Backend

Dans cette section, nous allons aborder la gestion des fonctionnalités et des opérations côté serveur, ainsi que la logique métier, l'interaction avec la base de données, la communication avec le frontend,

### 7.1.1 Base de données

La base de données PostgreSQL a été créée sur Google Cloud (GCP). Pour créer une base de données PostgreSQL sur Google Cloud Platform (GCP), tout d'abord, nous devons suivre les étapes suivantes :

- **Connectez-vous à votre compte GCP :** Assurez-vous que vous êtes connecté à votre compte GCP. Si vous n'avez pas de compte, vous devez en créer un.
- **Créez un projet (si nécessaire) :** Si vous n'avez pas déjà un projet GCP, vous pouvez en créer un en cliquant sur le bouton "Sélectionner un projet" en haut de la page, puis en cliquant sur "Nouveau projet".
- **Sélectionnez ou créez une instance de base de données SQL :** Dans la console GCP, cliquez sur "SQL" dans le menu de navigation à gauche.
- **Configurer les autorisations :** Dans l'onglet "Autorisations" de votre instance PostgreSQL, vous pouvez configurer les autorisations pour les utilisateurs et les réseaux qui auront accès à la base de données.
- **Créer une base de données :** Une fois votre instance PostgreSQL créée, vous pouvez créer une base de données en utilisant un client SQL tel que `psql` ou en utilisant la Console GCP.
  - a. En utilisant `psql` :
    - Pour créer une base de données en utilisant `psql`, vous pouvez vous connecter à votre instance PostgreSQL à l'aide de la commande `psql` en spécifiant le nom de la base de données par défaut, puis exécuter la commande SQL pour créer une nouvelle base de données. Par exemple :

```
psql -h [adresse_IP_instance] -U [utilisateur] -d postgres
```

```
CREATE DATABASE my_data_base
```

- b. En utilisant la Console GCP :

Dans la Console GCP, allez à l'onglet "Bases de données" de votre instance PostgreSQL, puis cliquez sur "Créer une base de données". Remplissez les détails de la base de données, tels que le nom et le jeu de caractères, puis cliquez sur "Créer".

Vous avez maintenant créé une base de données PostgreSQL sur Google Cloud Platform (GCP) et vous pouvez commencer à l'utiliser pour stocker vos données. Assurez-vous de bien sécuriser votre base de données en configurant les autorisations et en utilisant des pratiques de sécurité recommandées.

Et à la fin, ajoutez la configuration dans notre projet Django en local. La figure suivante montre les détails

```
100
109 DATABASES = {
110     'default': {
111         'ENGINE': 'django.db.backends.postgresql',
112         'NAME': 'khadematydev',
113         'USER': 'postgres',
114         'PASSWORD': '████████████████████████████████',
115         'HOST': '/cloudsql/khadematy-apidev:europe-west1:khadematy-inst-dev',
116         # 'HOST': '127.0.0.1',
117         'PORT': '5432',
118     }
119 }
120
121
```

Figure 18 : configuration GCP

### 7.1.2 Système d'envoie SMS by phone number

Dans notre application, les messages SMS sont un aspect essentiel à la fois pour la vérification de l'utilisation de la carte SIM et de son existence, ainsi que pour assurer la confidentialité.

Pour implémenter ce système, nous avons utilisé le package en django qui s'appelle HOTP

Le package HOTP (HMAC-Based One-Time Password) est un algorithme d'authentification à deux facteurs (2FA) qui génère des codes à usage unique (OTP) basés sur une clé secrète partagée entre le serveur et le client. Voici comment il fonctionne en fonctionnalité :

- **Configuration initiale** : Tout d'abord, un utilisateur ou un système configure l'authentification HOTP en enregistrant une clé secrète partagée avec le serveur d'authentification. Cette clé est stockée en toute sécurité et doit rester confidentielle.
- **Génération de codes OTP** : Lorsqu'un utilisateur souhaite s'authentifier, le client génère un code OTP en utilisant la clé secrète partagée et un compteur (ou une horloge). Le compteur est incrémenté à chaque utilisation pour garantir que chaque

code est unique. L'algorithme HMAC (Hash-based Message Authentication Code) est généralement utilisé pour générer le code OTP.

- **Validation côté serveur :** L'utilisateur soumet le code OTP au serveur d'authentification. Le serveur possède également la clé secrète et le compteur associé à cet utilisateur. Le serveur génère un code OTP en utilisant la même clé secrète et le même compteur, puis compare le code soumis par l'utilisateur avec le code généré. Si les deux codes correspondent, l'authentification est réussie.
- **Synchronisation du compteur :** Après chaque validation réussie, le serveur met à jour le compteur de l'utilisateur pour refléter la dernière utilisation. Cela garantit que le prochain code OTP généré par le client sera basé sur un compteur supérieur.
- **Utilisation des codes OTP :** Les codes OTP générés sont à usage unique, ce qui signifie qu'ils ne peuvent être utilisés qu'une seule fois et ont une durée de validité limitée. Cela renforce la sécurité de l'authentification, car même si un attaquant intercepte un code OTP, il ne sera pas valable pour une utilisation ultérieure.
- **Répétition du processus :** Ce processus est répété à chaque tentative d'authentification. À chaque connexion, le compteur est incrémenté, et un nouveau code OTP est généré.
- En résumé, HOTP est basé sur la génération de codes OTP à partir d'une clé secrète partagée et d'un compteur, avec une validation côté serveur pour s'assurer que le code soumis par l'utilisateur correspond à celui généré par le serveur. Cela renforce la sécurité de l'authentification en ajoutant un deuxième facteur (le code OTP) en plus du mot de passe traditionnel.

HOTP est largement utilisé pour l'authentification à deux facteurs dans divers contextes, notamment les systèmes de sécurité en ligne, les applications bancaires, et d'autres systèmes qui exigent une sécurité accrue.

### 7.1.3 Operations CRUD

Dans cette section, nous allons aborder les opérations CRUD (Create, Read, Update, Delete) présentes dans notre application. Les entités sur lesquelles nous allons nous concentrer sont les suivantes : Reference, abonnement, branchement, auto-releve,facture.

Pour toutes ces entités, les routes suivent généralement une structure similaire, avec des objectifs spécifiques pour chaque route.

Le tableau suivant résume ces routes ainsi que leur utilité et les méthodes HTTP qu'elles utilisent et entités concernées:

Tableau : Résumé des routes, utilité , entités concernées et méthodes HTTP utilisées

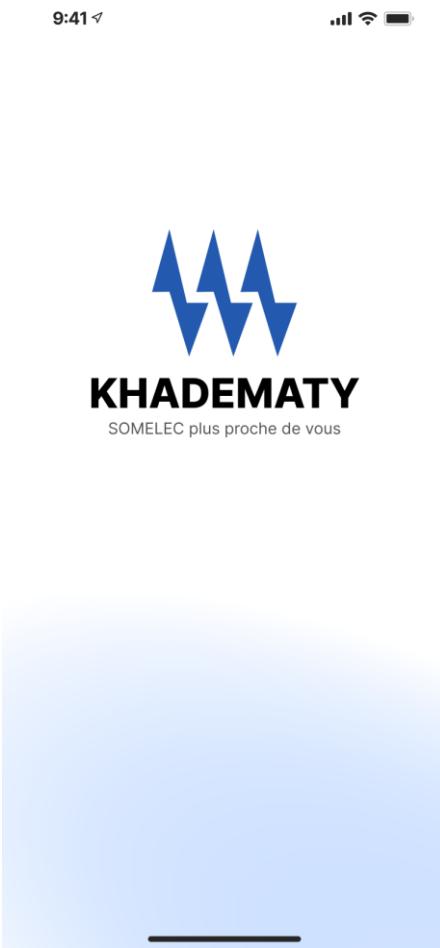
Route	Méthode HTTP	utilité	entités concernées
creation	POST	Permet de créer une nouvelle entité en fournissant les informations nécessaires.	Branchement Abonnement Reference Auto-releve
list	GET	Permet de lister tous les éléments d'une entité, offrant ainsi une vue globale de toutes les données disponibles.	Facture Reference Abonnement Branchement Auto-releve
Detail par id	GET	Permet de récupérer les détails d'un élément spécifique d'une entité en utilisant un identifiant unique (ID)	Facture Reference
Modification	PUT	Permet de mettre à jour les informations d'un élément d'une entité en utilisant son identifiant unique (ID)	Reference
Delete	DELETE	Permet de supprimer un élément d'une entité.	Reference

## 7.2 Frontend

La réalisation de ce travail s'est fait sur plusieurs étapes, partant de la mise en place des outils de développement, jusqu'au développement proprement dit des fonctionnalités du système.

Ce chapitre portera, en effet, sur la préparation de l'environnement de travail logiciel, matériel. Ainsi j'allais s'intéresser essentiellement à une description de différentes pages constitutants le site web ; en effet on va présenter tout d'abord l'architecture de site, ensuite les liens de navigations, puis la description des pages du site.

Dans cette section, nous allons aborder présenter l'ensemble des screens de l'application.



*Figure 19 : on splash screen*

Ce screen est la première page qu'il s'affiche à l'utilisateur il permet d'indiquer à l'utilisateur que l'application est en cours d'exécution, avec une brève description de l'application, telle que son nom et son icône.

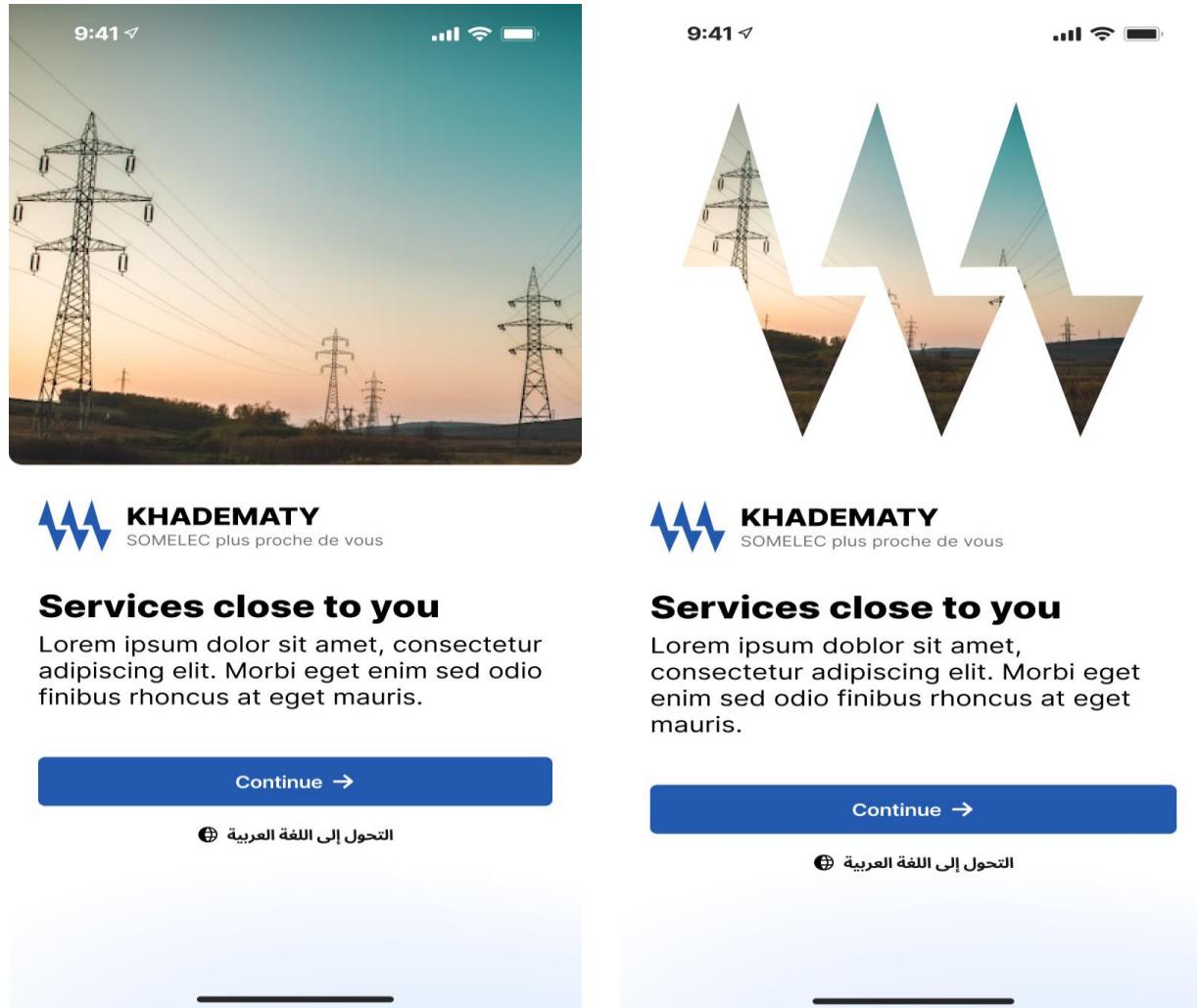


Figure 20 : on boarding screen

Les écrans d'onboarding (ou écrans de démarrage) sont utilisés dans les applications pour guider les nouveaux utilisateurs à travers les fonctionnalités et l'interface de l'application dès leur première utilisation. Leur rôle principal est de fournir une introduction et une explication sur la manière d'utiliser l'application.

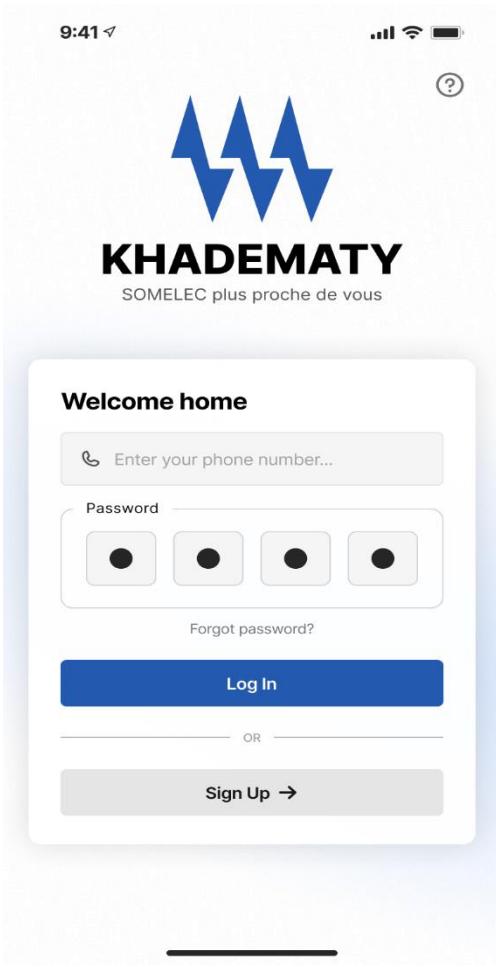
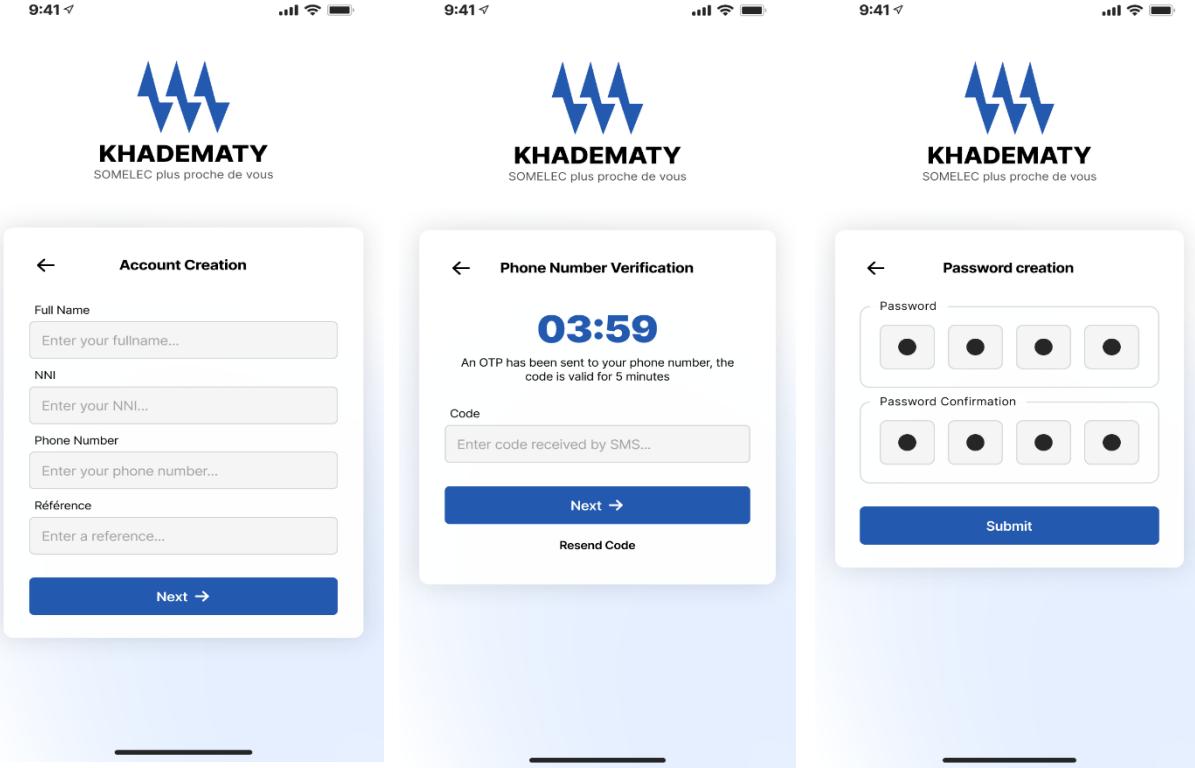


Figure 21 : login screen

Le rôle de cette partie de l'application (login screen) est de permettre aux utilisateurs d'entrer leurs informations d'identification (telles que le numéro d'utilisateur et le mot de passe) afin d'accéder à un système, une application ou une plateforme informatique.

Dans cette partie, nous avons utilisé l'authentification SIMPLE JSON Web Token (JWT) pour assurer une authentification très sécurisée et garantir que l'utilisateur ouvre une session sur une seule machine.



Cette fois-ci, le traitement permet à l'utilisateur de créer un compte dans la base de données via l'interface utilisateur en fournissant les informations de l'utilisateur. Ensuite, la validation ou la confirmation du code OTP est nécessaire pour confirmer que cet utilisateur est bien lui-même et le propriétaire de ce numéro de téléphone.

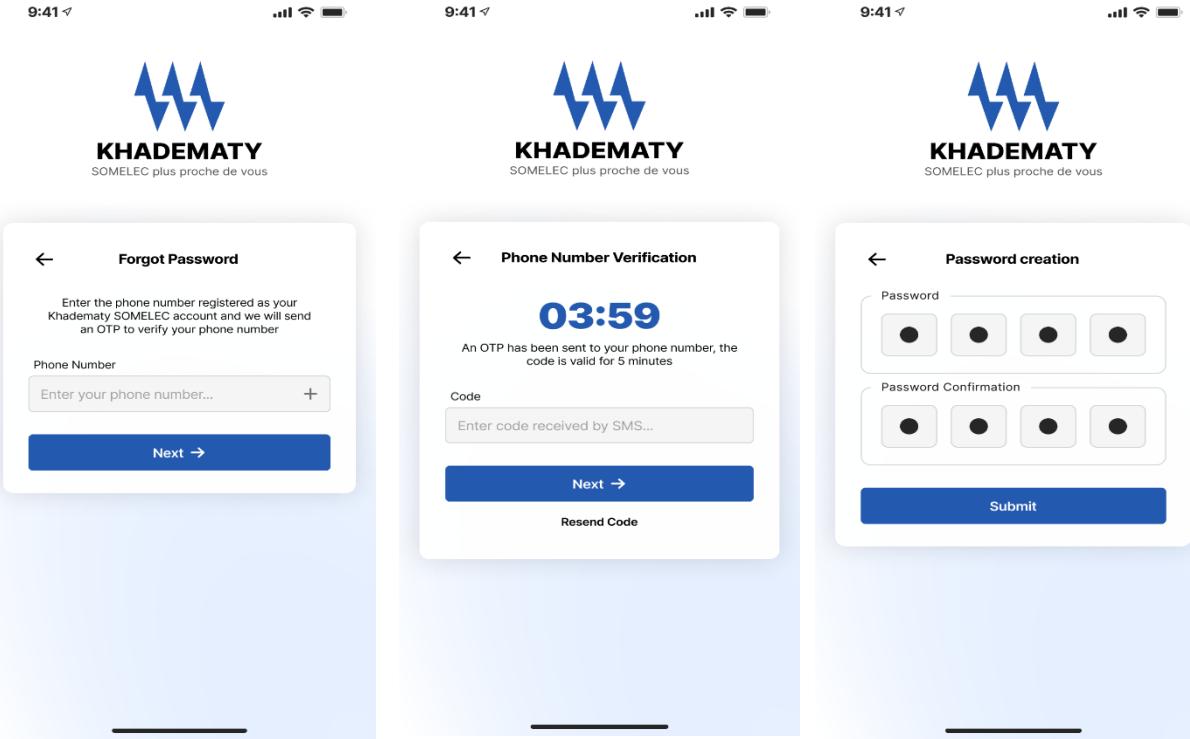


Figure 25 : forget password

L'objectif de cette étape est de donner à l'utilisateur la possibilité de changer ou récupérer son mot de passe. En passant par ces trois étapes, on renseigne d'abord le numéro de téléphone, puis on reçoit un message SMS sur notre numéro de téléphone pour confirmer que nous sommes le responsable de ce numéro de téléphone. Après, nous aurons la possibilité de changer le mot de passe.

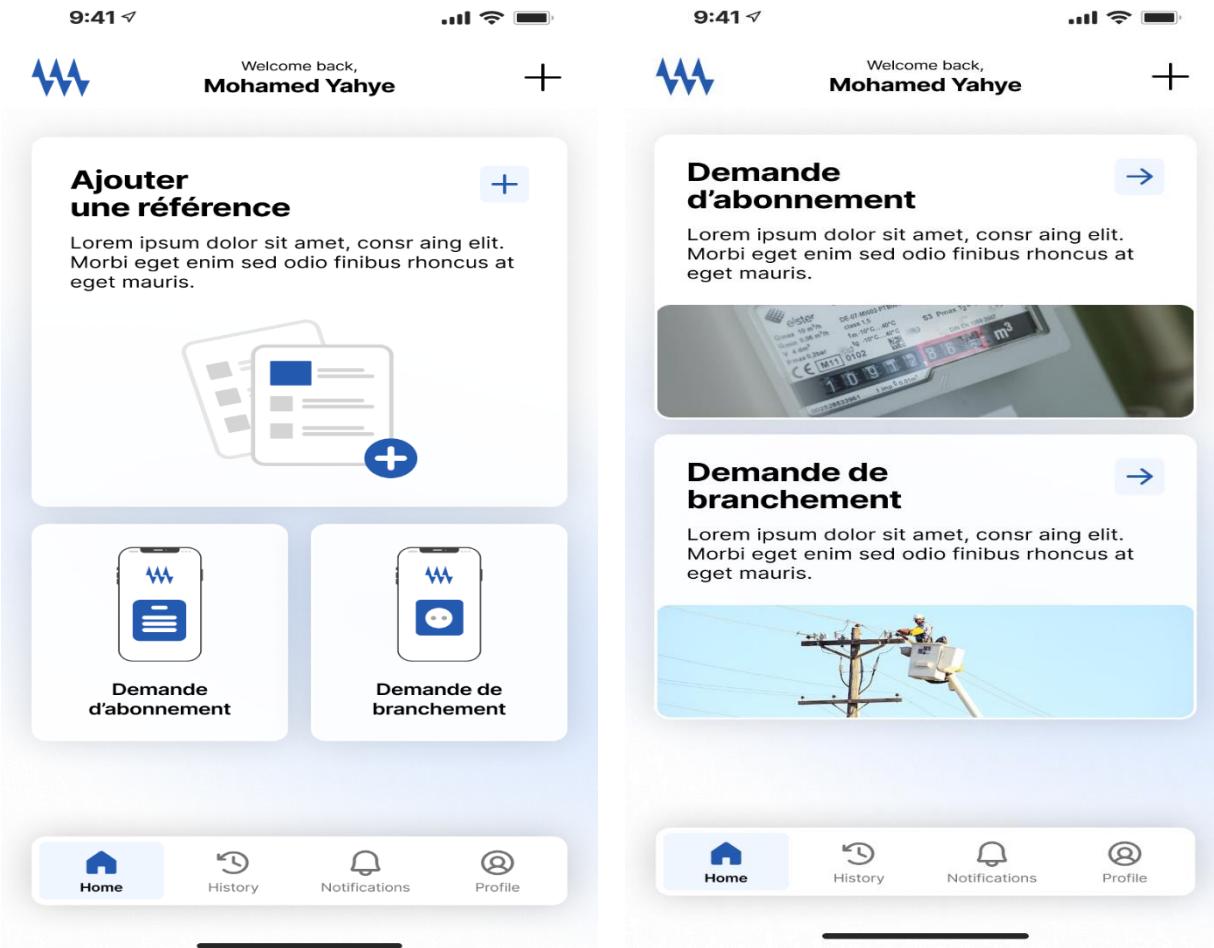


Figure 26: home screen (no ref)

C'est la première page qui s'affiche à l'utilisateur s'il est la première fois à installer l'application. Elle offre trois fonctionnalités : l'ajout de branchement, l'abonnement et la référence s'ils existent dans la base de données.

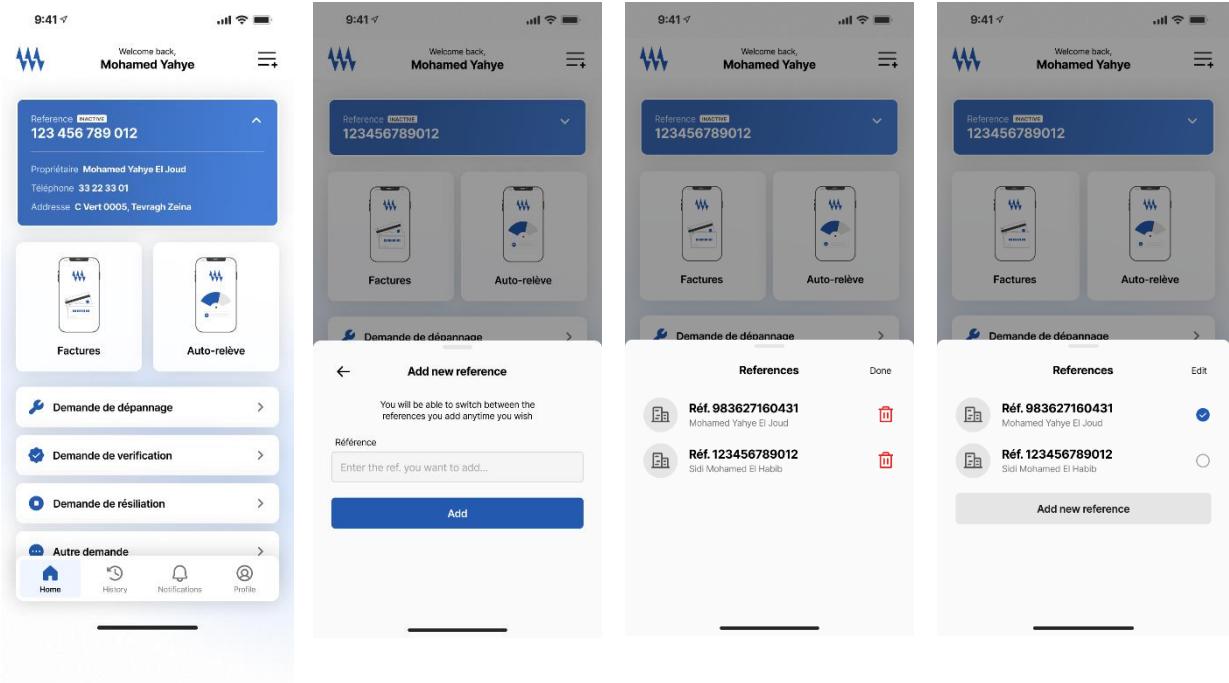


Figure 27 : home(réf)

S'il a déjà une référence, il peut consulter son solde, le nom du propriétaire, le numéro de téléphone, et l'adresse. De plus, il peut consulter ses factures et les télécharger, et il peut effectuer trois fonctionnalités : l'auto-relevé, la demande de dépannage, la demande de vérification, et la demande de résiliation

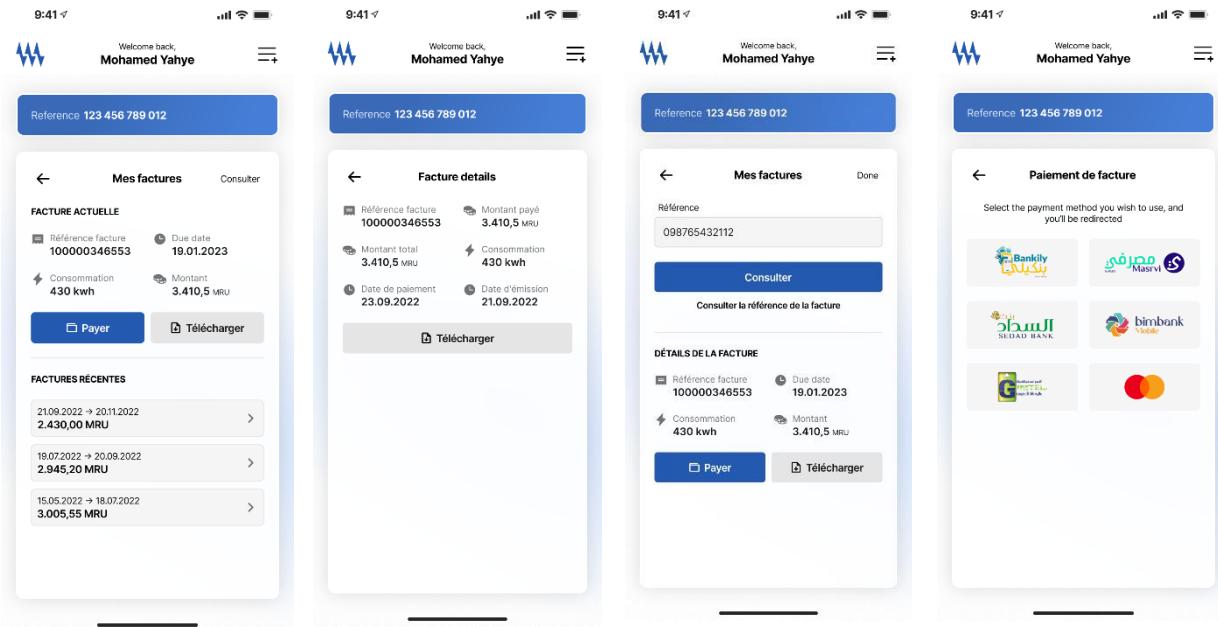


Figure 28 : facture

Cette partie de la facturation permet au client de lister ses factures et de pouvoir accéder aux détails de chaque facture. Il a la possibilité de les payer, de les télécharger, et de rechercher une facture par sa référence. Il peut également sélectionner l'un des portefeuilles (wallet) pour effectuer le paiement

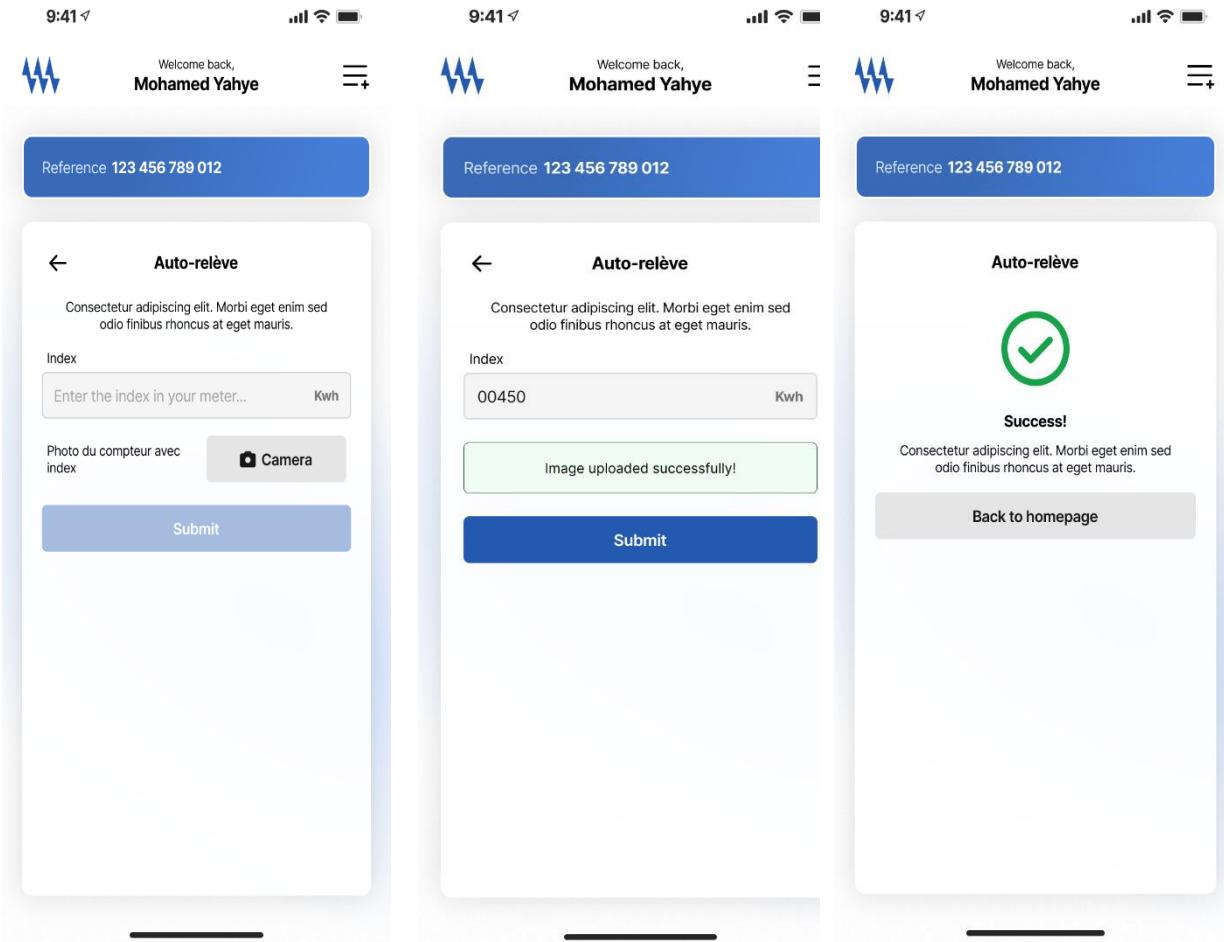


Figure 29 : Auto relève

Dans cette partie, il nous explique les étapes pour faire une demande d'auto-relevé de paiement. On ajoute l'index et on sélectionne l'image du compteur pour lequel nous souhaitons effectuer l'auto-relevé. Ensuite, nous envoyons la requête et attendons la réponse.

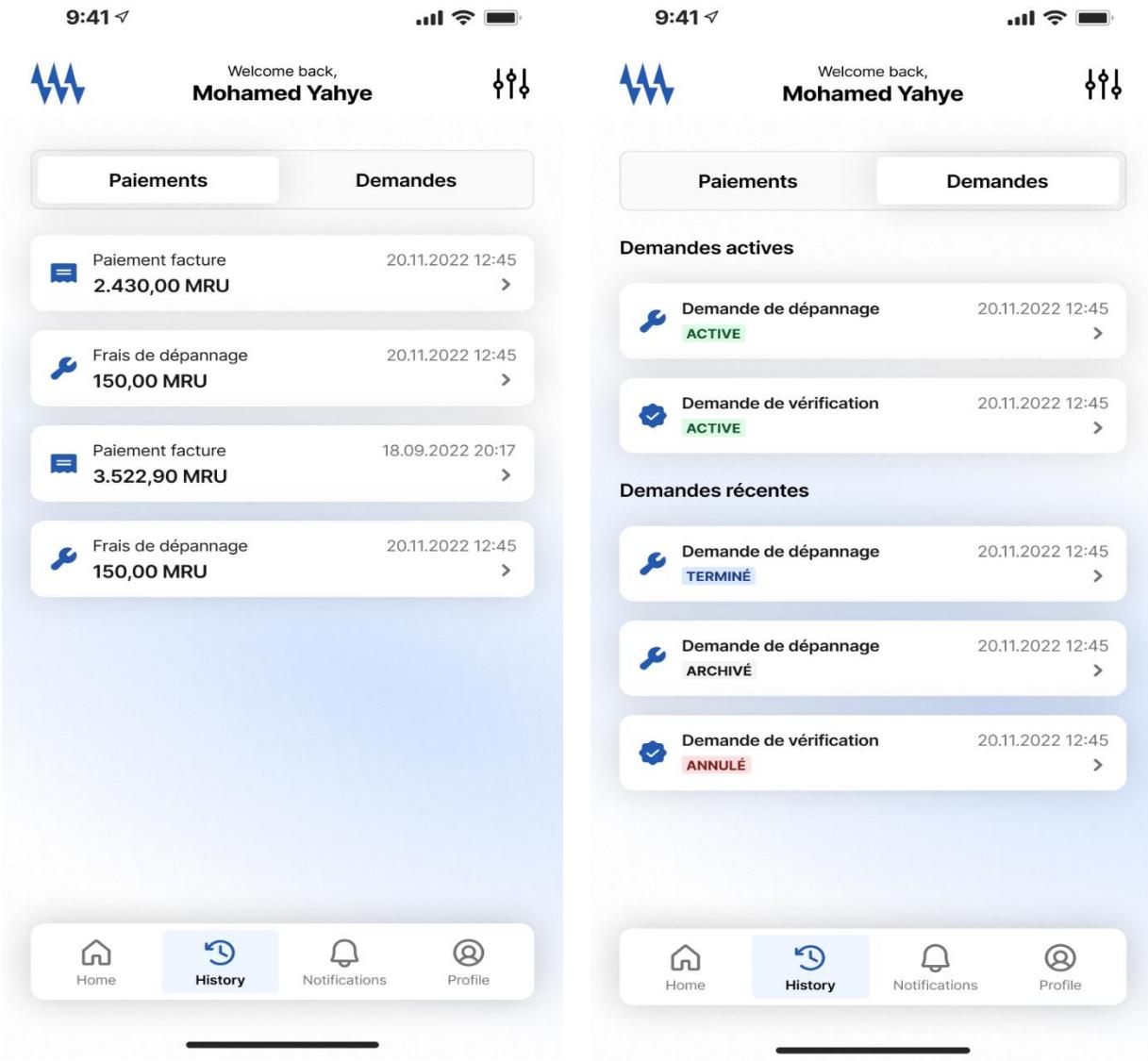


Figure 30 : historique

Cette page permet de répertorier toutes les demandes et les paiements effectués par l'utilisateur, quelle que soit leur nature. Elle comprend trois types de demandes : actives après l'activation et le traitement de la demande, annulées si la demande a été refusée, et terminées si la demande est achevée.

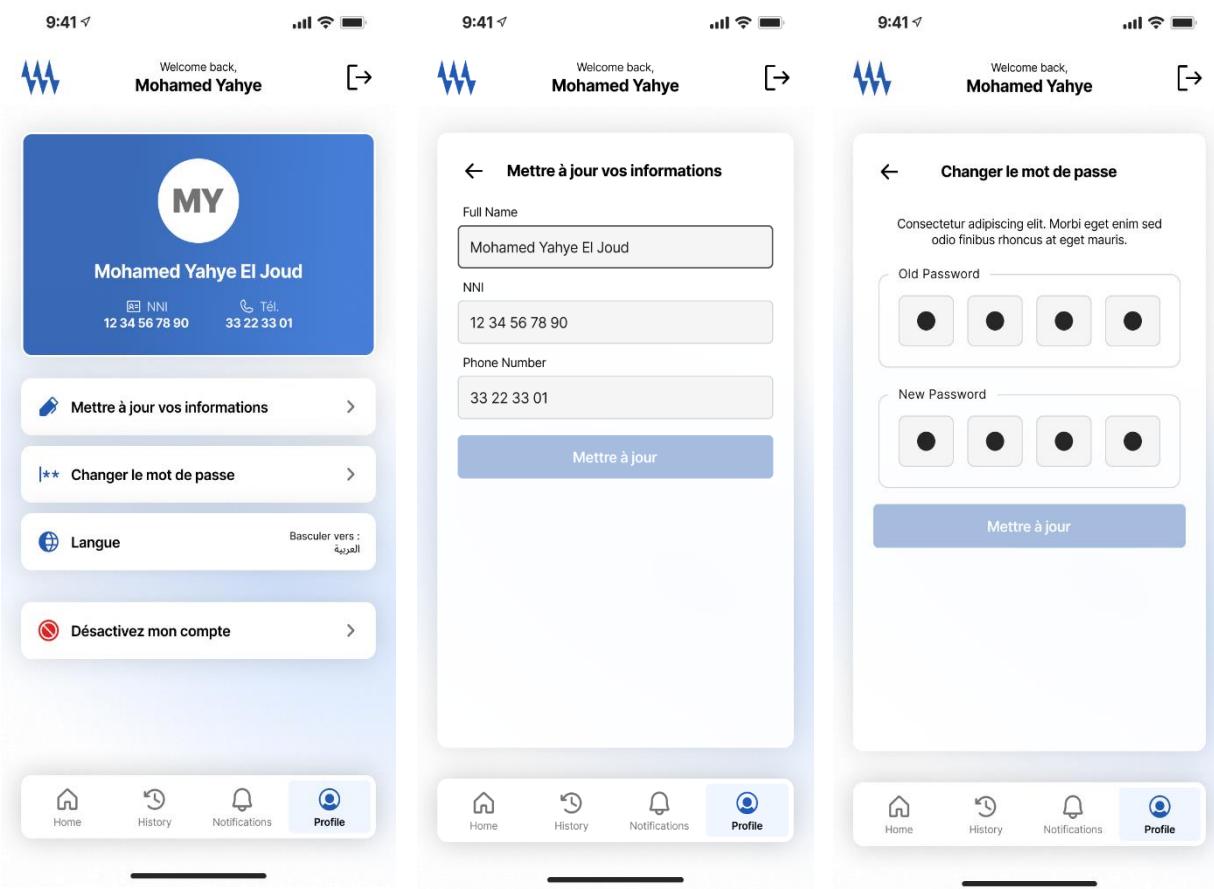


Figure 31 : profile screen

La page de profile comprend les informations de l'utilisateur de l'application, avec la possibilité de les modifier, ainsi que la possibilité de désactiver le compte. Il n'est pas possible de supprimer le compte, car l'entité utilisateur est liée à d'autres entités. Si l'une des entités est supprimée, l'autre entité sera automatiquement supprimée. C'est pourquoi nous utilisons la désactivation de compte.

# Conclusion générale

En conclusion, mon stage a été une expérience extrêmement enrichissante qui m'a permis d'acquérir de nouvelles compétences, de mettre en pratique mes connaissances académiques et de découvrir le monde professionnel. J'ai eu l'opportunité de travailler au sein d'une équipe dynamique et compétente, ce qui m'a permis d'apprendre énormément et de contribuer de manière significative aux projets de l'entreprise. enrichissante, m'offrant l'opportunité de : enrichissante

- Valider mon choix de carrière en informatique en affrontant de nouveaux défis professionnels.
- Évaluer ma capacité d'adaptation et de réaction en milieu professionnel.
- Acquérir une maîtrise complète des aspects du développement et de la conception, couvrant
- l'ensemble du spectre de la programmation.
- Consacrer du temps à la consolidation de mes connaissances, notamment en me perfectionnant
- Dans les domaines de la programmation mobile et du développement backend, étant donné la prévalence croissante de ces compétences au sein des entreprises en raison de leurs nombreux avantages.

Au cours de cette expérience, j'ai investi beaucoup d'efforts pour acquérir une expertise approfondie dans des domaines tels que Java, Python, Django, le développement Agile (Scrum), l'utilisation de Jira, ainsi que les technologies React et Flutter.

# Références

**Site Web:**

- 1- <https://docs.djangoproject.com/en/4.2/>
- 2- <https://django-rest-framework-simplejwt.readthedocs.io/en/latest/>
- 3- <https://pub.dev/>
- 4- <https://chat.openai.com/c/299ebce2-366a-4ae2-afc1-2acd9de41aa7>

**Livre:**

5. Django for Professionals, William S. Vincent
6. High Performance PostgreSQL
7. Le guide de l'utilisateur UML, James Rumbaugh, Grady Booch
6. Clean Code