



Rapport de stage d'initiation

Présenté et réalisé par :

Masmoudi Iheb

Application web pour enregistrer les compteurs d'électricité, de gaz et d'eau pour suivre la consommation

Préparé au sein de :

Société ETAP



Durée : du 1 juillet jusqu'à 11 août

Encadrant professionnel :

Mohamed ksibi

Année Universitaire : 2022/2023



Table des matières

Introduction Générale	1
1 Context général	2
1.1 Contexte et objectifs	3
1.2 Présentation de l'organisme d'accueil	3
1.3 Présentation de l'établissement	4
1.4 Objectif du projet	5
1.4.1 Problématique	5
1.4.2 Solution	5
1.5 Langage et méthodologie de modélisation	6
1.5.1 Méthodologie	6
2 Etude et planification de l'application	7
2.1 Spécification des besoins fonctionnels et non fonctionnels	8
2.1.1 Besoins fonctionnels	8
2.1.2 Besoins non fonctionnels	9
2.2 Spécification des acteurs	9
2.3 Diagramme de cas d'utilisation Générale	10
2.4 Diagramme de cas d'utilisation raffiné	12
2.4.1 Diagramme de cas d'utilisation « gestion des paramètres »	12
2.4.2 Diagramme de cas d'utilisation « Gestion des factures»	13
2.4.3 Diagramme de cas d'utilisation «gestion des utilisateurs»	14
2.5 Diagramme de classe d'analyse	15
2.6 Le model MVC	16
2.7 Les prototypes des interfaces de l'application	17
3 Conception	22
3.1 Diagrammes de séquence objets	23
3.1.1 Diagrammes de séquence objets « Authentification »	23
3.1.2 Diagrammes de séquence objets « Gestion facture »	24

3.1.3	Diagrammes de séquence objets « Gestion utilisateur »	25
3.2	Diagramme de classe de conception	26
4	RÉALISATION	27
4.1	Outils de réalisation	28
4.2	Réalisation	30
4.2.1	login	30
4.2.2	Page initial	30
4.2.3	Gestion région	31
4.2.4	Gestion locale	31
4.2.5	Gestion emplacements	32
4.2.6	Gestion utilisateurs	32
4.2.7	Gestion compteurs	33
4.2.8	Gestion types de compteurs	33
4.2.9	Interface facture	34
	Conclusion générale	35

Table des figures

1.1	Logo de la société	3
1.2	Logo de l'établissement	4
1.3	l'approche de la méthode waterfall	6
2.1	les acteurs de l'application	9
2.2	Diagramme de cas d'utilisation générale	10
2.3	Diagramme de cas d'utilisation «Admin - Gestion des paramètres»	12
2.4	diagramme de cas d'utilisation « Gestion des factures»	13
2.5	Diagramme de cas d'utilisation « Gestion des utilisateurs»	14
2.6	Diagramme de classe d'analyse	15
2.7	Model MVC vs micro-services	16
2.8	Interface « login »	17
2.9	Interface « Home page »	17
2.10	Interface « Gestion factures »	18
2.11	Interface « Ajouter facture »	18
2.12	Interface « Imprimer facture »	19
2.13	Interface « Gestions des compteurs»	19
2.14	Interface « Gestions des utilisateurs »	20
2.15	Interface « Gestions des rôles »	20
2.16	Interface « Gestions des locales »	21
3.1	Diagramme de cas d'utilisation «Admin - Authentification»	23
3.2	Diagramme de cas d'utilisation «Admin - Gestion facture»	24
3.3	Diagramme de cas d'utilisation «Admin - Gestion utilisateur»	25
3.4	Diagramme de classe de conception	26
4.1	visual paradigm	28
4.2	Laravel	28
4.3	XAMPP	28
4.4	Visual Studio Code	29

4.5	PhpMyAdmin	29
4.6	GitHub	29
4.7	Réalisation de la page login	30
4.8	Réalisation de la page initial	30
4.9	Réalisation « Gestion région »	31
4.10	Réalisation « Gestion locale »	31
4.11	Réalisation « Gestion emplacements »	32
4.12	Réalisation « Gestion cours »	32
4.13	Réalisation « Gestion compteurs »	33
4.14	Réalisation « Gestion types de compteurs »	33
4.15	Réalisation « Gestion facture »	34

Liste des tableaux

2.1	Description textuelle du rôle de l'admin	8
2.2	Description textuelle du rôle de l'utilisateur	8
2.3	Description textuelle du rôle du super admin	11
2.4	Description textuelle du diagramme de cas d'utilisation du « Admin – authentification»	13
2.5	Description textuelle du diagramme de cas d'utilisation du «Admin - Gestion des paramètres»	13
2.6	Description textuelle du diagramme de cas d'utilisation de « Gestion des factures» .	14
2.7	Description textuelle du diagramme de cas d'utilisation du « Gestion des utilisateurs»	15

Liste des abréviations

- **API** = Application Programming Interface
- **JSON** = JavaScript Object Notation
- **UML** = Unified Modeling Language

Introduction Générale

La gestion précise de la consommation d'électricité, de gaz et d'eau est d'une importance vitale pour les entreprises soucieuses de contrôler leurs coûts énergétiques et de promouvoir une utilisation responsable des ressources. Dans le cadre de ce projet, l'entreprise ETAP reconnaît la nécessité d'une solution efficace pour enregistrer les relevés de compteurs, suivre la consommation et gérer les factures correspondantes. Ainsi, le développement d'une application web dédiée à cet objectif se révèle essentiel.

Cette application permettra aux utilisateurs d'accéder facilement à toutes les informations pertinentes concernant la consommation d'électricité, de gaz et d'eau, tout en facilitant la gestion des factures. Les fonctionnalités incluront la gestion des caractéristiques générales du système, la saisie des relevés de compteurs, le calcul précis de la consommation et la génération de factures personnalisées.

Ce rapport désigne l'ensemble des étapes suivies pour développer la solution. Il contient 4 chapitres organisés. Le premier chapitre sera consacré à la présentation de l'organisme où nous allons présenter le cadre du projet ainsi que l'organisme d'accueil on doit aussi analyser la critique de l'existence en spécifiant les problèmes et leurs solutions. Nous allons expliquer à la fin notre méthodologie utilisée.

Dans le deuxième chapitre nous spécifierons les besoins fonctionnelles, non fonctionnelles et les acteurs de l'application. Nous conceptualiserons les besoins de l'utilisateur en utilisant plusieurs diagrammes cas d'utilisation , ensuite nous allons mettre en évidence l'architecture de l'application puisque nous avons utilisé une architecture spécifique en élaborant les prototypes et les outils utilisés.

Le troisième chapitre « Conception » expliquera plus de détails sur le fonctionnement de l'application avec les diagrammes de séquence objets et le diagramme de classe de conception.

Le quatrième chapitre « Implémentation et tests » présentera la réalisation finale des interfaces. Pour finir, avoir une vue générale avec une conclusion finale de tout le rapport.

CONTEXT GÉNÉRAL

Plan

1	Contexte et objectifs	3
2	Présentation de l'organisme d'accueil	3
3	Présentation de l'établissement	4
4	Objectif du projet	5
5	Langage et méthodologie de modélisation	6

Introduction

Ce chapitre sera consacré à l'aperçu général de notre projet durant notre stage d'été et à la présentation de l'organisme d'accueil. Ensuite, des problématiques et des solutions possibles seront discutées. La dernière section présentera la méthodologie.

1.1 Contexte et objectifs

Notre projet de stage d'initiation consiste à développer une application web pour enregistrer les compteurs d'électricité, de gaz et d'eau pour suivre la consommation au sein de l'entreprise ETAP. L'objectif principal de cette application est de simplifier et d'améliorer le processus de suivi des consommations dans chaque région.

1.2 Présentation de l'organisme d'accueil

ETAP (Entreprise Tunisienne d'Activités Pétrolières) est une société tunisienne spécialisée dans l'exploration, la production et la commercialisation des hydrocarbures. Fondée en 1972, elle est l'entreprise nationale du secteur pétrolier en Tunisie. **figure 1.1[1]**.



FIGURE 1.1 : Logo de la société

En tant que pilier de l'industrie pétrolière tunisienne, l'ETAP contribue de manière significative à l'économie nationale en fournissant des emplois, en générant des revenus fiscaux et en assurant l'approvisionnement en énergie du pays.

Le service informatique nous a accueillis durant notre formation. Nous y effectuons notre stage d'initiation afin de réaliser une application web pour enregistrer les compteurs d'électricité, de gaz et d'eau pour suivre la consommation.

1.3 Présentation de l'établissement



FIGURE 1.2 : Logo de l'établissement

Polytechnique est une école d'enseignement supérieur créée en 2013 privé par le ministère de l'enseignement Supérieur sous l'accord 04/2013 qui cible la formations hauts cadres techniques présenter par le logo de la **figure 1.2[2]**.

Les spécialités :

- **Architecture :**

- Diplome National d'architecture
- Master en Building Information Modeling (BIM)

- **Cycle préparatoire :**

- Ingénieur Système Embarqué et Mobiles
- Ingénieur Big Data Business Intelligence
- Ingénieur IT finance
- Ingénieur Génie Industriel
- Ingénieur Mécatronique
- Ingénieur Système d'information et du logiciel
- Ingénieur Informatique , Réseaux et Multimédia

- **Cycle Ingénieur :**

- Ingénieur Système Embarqué et Mobiles

- **Licences :**

- Licence en informatique

1.4 Objectif du projet

Le projet consiste à développer une application web pour enregistrer les compteurs d'électricité, de gaz et d'eau pour suivre la consommation de l'entreprise ETAP. Cette application permettra aux utilisateurs de gérer les caractéristiques générales du système , gérer les factures de consommation de gaz naturel et de gérer les factures de consommation d'électricité Par conséquent, la gestion des factures sera plus facile.

1.4.1 Problématique

- Quelles sont les principales fonctionnalités requises pour l'application web permettant d'enregistrer les compteurs d'électricité, de gaz et d'eau de l'entreprise ETAP et de suivre la consommation, tout en gérant les factures de consommation de gaz naturel et d'électricité ?
- Quels sont les défis techniques auxquels l'équipe de développement pourrait être confrontée lors de la création de cette application web pour assurer une gestion efficace des factures de consommation d'énergie pour l'entreprise ETAP ?
- Comment l'application web prévoit-elle de garantir la sécurité des données sensibles, telles que les informations de consommation et les données de facturation, tout en permettant aux utilisateurs autorisés de gérer les caractéristiques générales du système et d'accéder aux informations nécessaires pour une gestion optimale des factures ?

1.4.2 Solution

Utilisation d'une base de données centralisée : Mettre en place une base de données centralisée robuste permettant de stocker les relevés de compteurs, les données de consommation et les informations de facturation de manière sécurisée. Cette solution facilitera la gestion des factures en permettant un accès rapide et précis aux données nécessaires.

Interface conviviale et fonctionnalités avancées : Développer une interface utilisateur conviviale et intuitive, offrant aux utilisateurs de l'application la possibilité de gérer les caractéristiques générales du système, de visualiser les données de consommation et de générer des rapports personnalisés. En intégrant des fonctionnalités avancées telles que la prévision de la consommation future et l'analyse des tendances, l'application facilitera la gestion proactive des factures et l'identification d'éventuelles anomalies de consommation.

1.5 Langage et méthodologie de modélisation

1.5.1 Méthodologie

Pour les exigences bien définies de notre projet nous avons choisi le modèle waterfall de développement de logiciels linéaire. Dans lequel il suit un processus en cascade, où chaque étape dépend de la précédente et prépare la suivante. Le modèle en cascade reposant sur les exigences de la division des processus de développement en cinq phases de projet simplifiés par la **figure 1.3** [3].

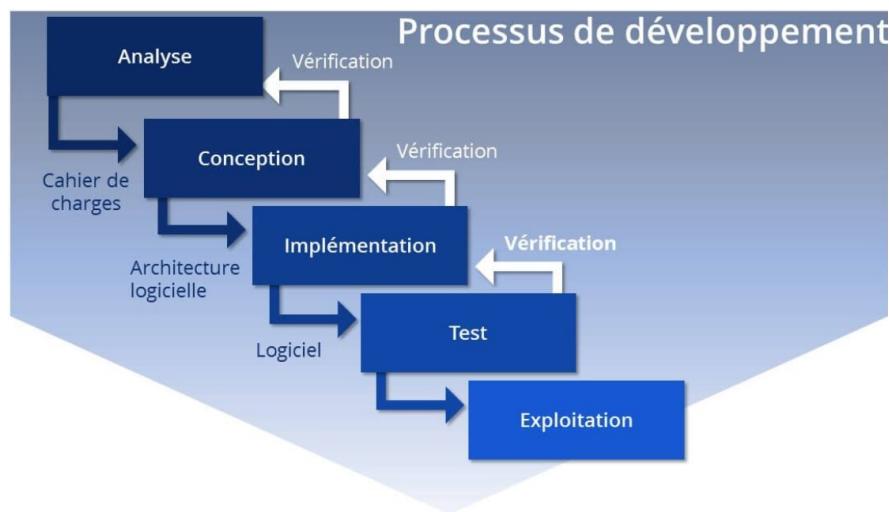


FIGURE 1.3 : l'approche de la méthode waterfall

Les étapes du modèle Waterfall incluent l'analyse des besoins, la conception, le développement, les tests, la mise en œuvre et la maintenance. Pour mieux expliquer chaque phases :

- **Analyse des besoins** : dans cette phase, les besoins du client sont identifiés et documentés.
- **Conception** : cette phase consiste à concevoir l'architecture du système, à identifier les composants et les interactions nécessaires et à créer un plan de développement.
- **Développement** : le code du système est écrit et les composants sont développés.
- **Tests** : le système est testé pour s'assurer qu'il répond aux exigences et qu'il fonctionne correctement.
- **Exploitation** : après la validation des tests, le système est déployé et livré au client.

Conclusion

En premier lieu nous avons choisi de présenter notre école en expliquant dans un deuxième lieu le cadre de notre projet et les critiqués qu'on a étudié pour comprendre les besoins du projet.

ETUDE ET PLANIFICATION DE L'APPLICATION

Plan

1	Spécification des besoins fonctionnels et non fonctionnels	8
2	Spécification des acteurs	9
3	Diagramme de cas d'utilisation Générale	10
4	Diagramme de cas d'utilisation raffiné	12
5	Diagramme de classe d'analyse	15
6	Le model MVC	16
7	Les prototypes des interfaces de l'application	17

Introduction

Ce chapitre fera signe au besoins fonctionnelles et non fonctionnelles en spécifiant les acteurs et la planification des sprints. Nous allons présenter dans ce chapitre les diagrammes de cas d'utilisation en utilisant le langage UML nous élaborerons aussi des prototypes pour une vue plus visible avec les outils de réalisation.

2.1 Spécification des besoins fonctionnels et non fonctionnels

2.1.1 Besoins fonctionnels

TABLEAU 2.1 : Description textuelle du rôle de l'admin

Super admin	Description
S'authentifier	L'admin accède à l'application avec un login et un mot de passe.
Gérer les paramètres	L'admin est la seule personne qui a le droit de ajouter , consulter , supprimer et chercher les régions , types de compteurs , compteurs , emplacements et locales.
Gérer les factures	L'admin peut gérer les listes des factures , les factures payées , impayées et autres ou il peut ajouter , modifier ,imprimé et consulter les images des factures.
Gérer les utilisateurs	L'admin peut gérer la liste et les permissions des utilisateurs ou il peut ajouter , modifier , supprimer et afficher les permissions des utilisateurs.
Recevoir notifications	L'admin peut marquer les messages comme lue.
Trier le diagramme	L'admin peut trier le diagramme de la page d'accueil.

Ce tableau nous permet de voir les fonctionnalités de l'admin

TABLEAU 2.2 : Description textuelle du rôle de l'utilisateur

Utilisateur	Description
S'authentifier	L'utilisateur accède à l'application avec un login et un mot de passe.
Gérer les factures	L'admin peut gérer les listes des factures , les factures payées , impayées et autres ou il peut ajouter , modifier ,imprimé et consulter les images des factures.
Recevoir notifications	L'admin peut marquer les messages comme lue.
Trier le diagramme	L'admin peut trier le diagramme de la page d'accueil.

Ce tableau nous permet de voir les fonctionnalités de l'utilisateur

2.1.2 Besoins non fonctionnels

Les besoins non fonctionnels représentent les contraintes liées à l'implémentation et l'environnement internes du système aussi la fiabilité et la performance.

- **Sécurité** : L'application doit fournir la confidentialité de l'accès à la base de données de la société.
- **Performance** : Un temps de réponse et chargement de l'application acceptable lors de l'utilisation.
- **Fiabilité** : L'application doit fonctionner correctement.
- **Ergonomie** : L'application doit offrir une interface dynamique et jolie.
- **Maintenance** : Le code doit être clair pour pouvoir faire des futures évolutions ou améliorations.

2.2 Spécification des acteurs

Il y a deux acteurs qui interagissent avec l'application développée indiqué par **figure 2.1** :

Admin : Il a accès à toutes les fonctionnalités, il peut contrôler la gestion des utilisateurs, des factures , des paramètres .Il peut aussi recevoir des notifications et trier le diagramme.

Utilisateur : Il peut seulement s'authentifier et gérer ses factures. il peut aussi recevoir des notifications et trier le diagramme.



FIGURE 2.1 : les acteurs de l'application

2.3 Diagramme de cas d'utilisation Générale

Les diagrammes de cas d'utilisation qui sont des diagrammes UML ont été choisi pour représenter le comportement fonctionnel du système logiciel, **Figure 2.2.**

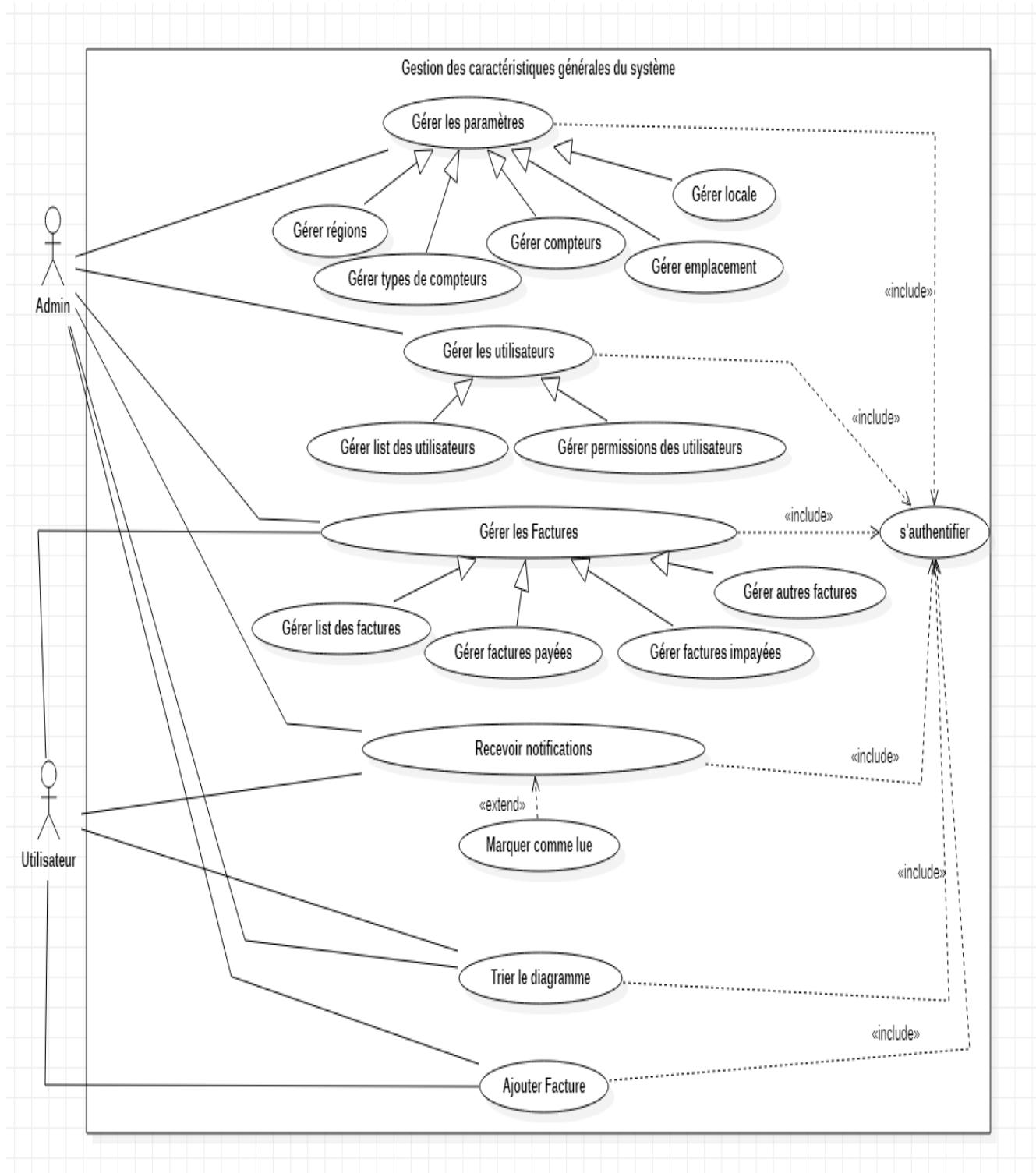


FIGURE 2.2 : Diagramme de cas d'utilisation générale

TABLEAU 2.3 : Description textuelle du rôle du super admin

Cas d'utilisation	Diagramme de cas d'utilisation générale
Acteur principal	Admin, utilisateur.
Parties prenantes et intérêts	Pour accéder à la page d'accueil les utilisateurs doivent insérer leurs emails et leurs mots de passe.
Précondition(s)	Authentification obligatoire de tous les acteurs.
Postcondition (s)	-----
Scénario principal	<p>L'admin peut :</p> <ul style="list-style-type: none"> - Trier le diagramme selon date debut et date fin et localisation. - Gérer les paramètres : région , type de compteur , compteur , emplacement et locale. - Gérer les factures : liste facture , facture payées , facture impayées et d'autres. - Gérer les utilisateurs : liste des utilisateurs et leurs permissions. - Ajouter des factures. - Recevoir des notifications qui peuvent être marquées comme lues. <p>L'utilisateur peut :</p> <ul style="list-style-type: none"> - Gérer les factures : liste facture , facture payées , facture impayées et d'autres. - Ajouter des factures. - Recevoir des notifications qui peuvent être marquées comme lues.
Scénario(s) alternatif(s)	Erreur de saisie

Ce tableau présente une description sur les fonctionnalités décrite dans le diagramme de cas d'utilisation générale.

2.4 Diagramme de cas d'utilisation raffiné

2.4.1 Diagramme de cas d'utilisation « gestion des paramètres »

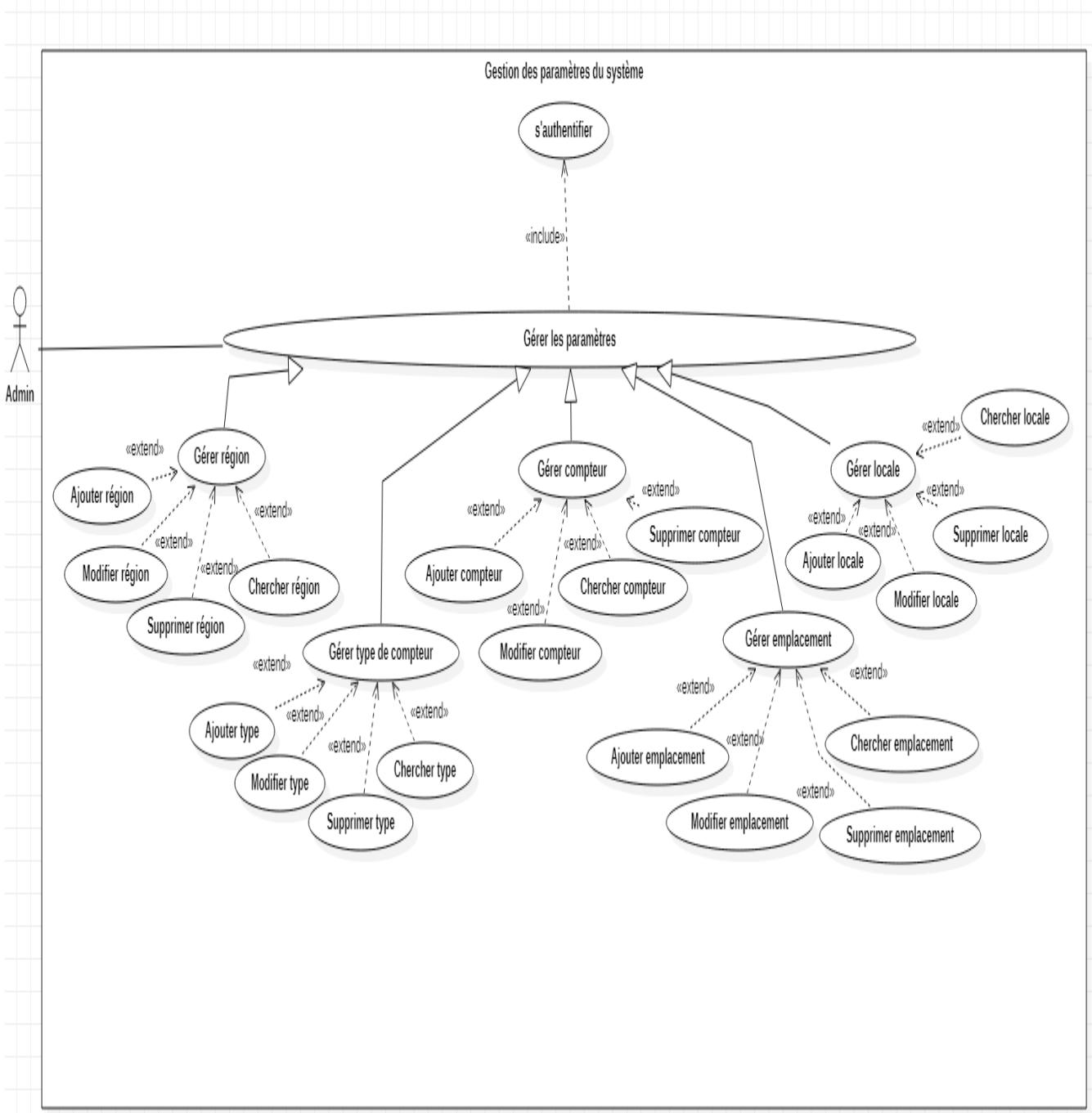


FIGURE 2.3 : Diagramme de cas d'utilisation «Admin - Gestion des paramètres»

Ce tableau présente une description du diagramme de cas d'utilisation de la fonction « S'authentifier »

TABLEAU 2.4 : Description textuelle du diagramme de cas d'utilisation du « Admin – authentification»

Cas d'utilisation	S'authentifier
acteur principal	Admin
Parties prenantes et intérêts	L'admin doit insérer son email et son mot de passe pour accéder au dashboard.
Précondition(s)	L'admin doit insérer son email.
Postcondition (s)	Passage à la page d'accueil.
Scénario principal	- L'admin peut saisir l'email et mot de passe. - L'affichage du dashboard.
Scénario(s) alternatif(s)	Erreur de saisie du mot de passe

TABLEAU 2.5 : Description textuelle du diagramme de cas d'utilisation du «Admin - Gestion des paramètres»

Cas d'utilisation	Gérer les paramètres
Acteur principal	Admin
Parties prenantes et intérêts	L'admin peut gérer les paramètres.
Précondition(s)	L'admin est tenu de s'authentifier.
Postcondition (s)	L'opération de saisie , modification , recherche ou de suppression est couronnée de succès.
Scénario principal	-L'admin peut ajouter, modifier , chercher ou supprimer des régions , des types de compteurs , des compteurs , des emplacements et des locales.
Scénario(s) alternatif(s)	Pas de paramètres disponible.

Ce tableau présente une description du diagramme de cas d'utilisation de la fonction « Gestion des paramètres »

2.4.2 Diagramme de cas d'utilisation « Gestion des factures»

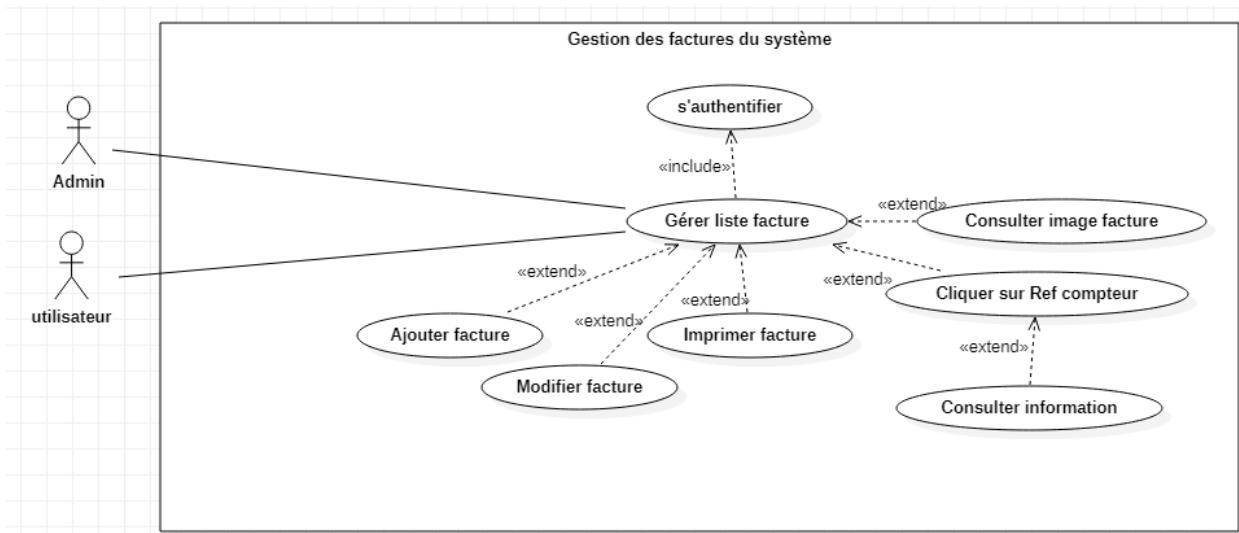


FIGURE 2.4 : diagramme de cas d'utilisation « Gestion des factures»

TABLEAU 2.6 : Description textuelle du diagramme de cas d'utilisation de « Gestion des factures»

Cas d'utilisation	Gestion des factures
acteur principal	Admin , utilisateur
Parties prenantes et intérêts	Pour accéder à la page d'accueil les utilisateurs doivent insérer leurs emails et leurs mots de passe.
Précondition(s)	L'utilisateur est tenu de s'authentifier.
Postcondition (s)	Gérer d'autre chose.
Scénario principal	-L'admin et l'utilisateur peuvent gérer les factures : ajouter , modifier imprimer et consulter l'image de facture. - pour accéder au information de la facture l'utilisateur doit cliquer sur la références du compteur
Scénario(s) alternatif(s)	-----

Ce tableau présente une description du diagramme de cas d'utilisation de la fonction « consulter le paramétrage de l'utilisateur »

2.4.3 Diagramme de cas d'utilisation «gestion des utilisateurs»

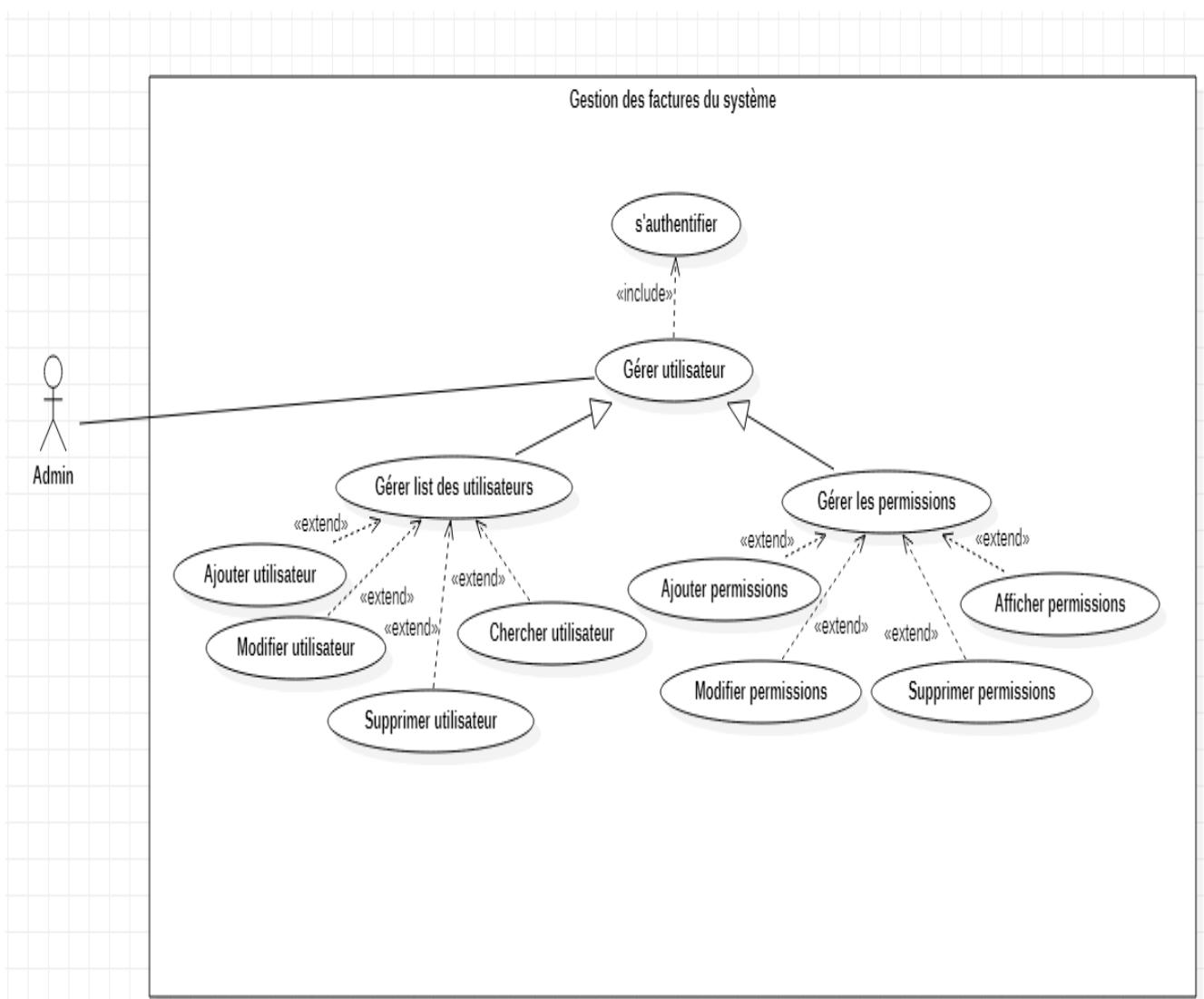


FIGURE 2.5 : Diagramme de cas d'utilisation « Gestion des utilisateurs»

TABLEAU 2.7 : Description textuelle du diagramme de cas d'utilisation du « Gestion des utilisateurs»

Cas d'utilisation	Gérer les utilisateurs
Acteur principal	Admin
Parties prenantes et intérêts	L'admin peut ajouter, modifier ou supprimer les utilisateurs.
Précondition(s)	L'admin est tenu de s'authentifier.
Postcondition (s)	L'opération de saisie , modification ou de suppression est couronnée de succès.
Scénario principal	-L'admin choisi de gérer les utilisateurs il peut donc ajouter, modifier , chercher ou supprimer les utilisateurs dans cette application. -Le responsable choisi de gérer les permissions il peut donc ajouter, modifier ou supprimer les permissions des utilisateurs dans cette application.
Scénario(s) alternatif(s)	Pas de utilisateurs disponible.

Ce tableau présente une description du diagramme de cas d'utilisation de la fonction « Gestion des utilisateurs»

2.5 Diagramme de classe d'analyse

Le diagramme de classes d'analyse permet de définir toute la structure conceptuelle d'une application et d'en faire ressortir toutes les contraintes ou règles de gestion.

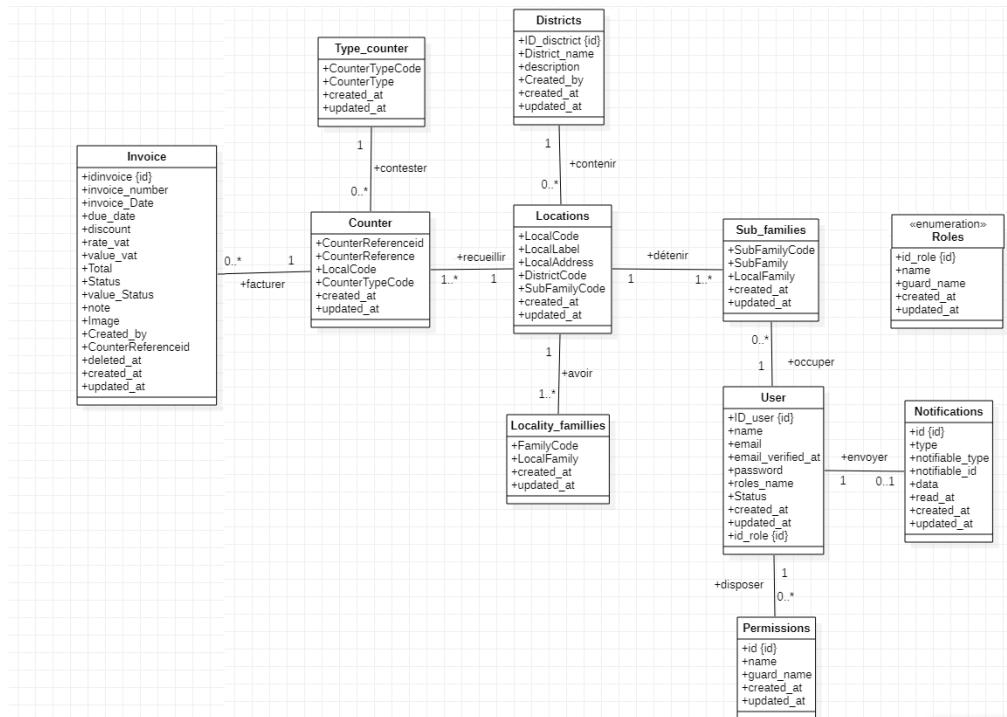


FIGURE 2.6 : Diagramme de classe d'analyse

2.6 Le model MVC

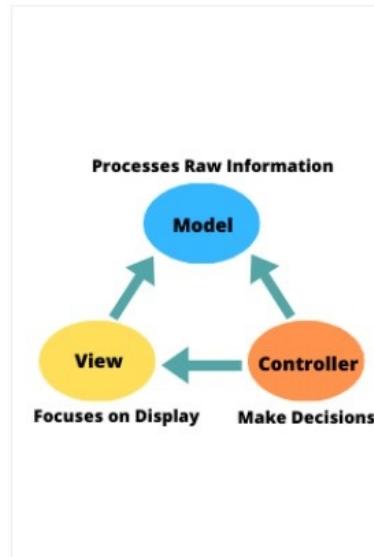


FIGURE 2.7 : Model MVC vs micro-services

Le model est un motif d'architecture découpée qui est largement utilisé pour concevoir et organiser les applications web. L'utilisateur interagisse avec la vue pour pouvoir récupérer la requête. Le contrôleur va définir la logique d'affichage, et afficher la vue suivante à l'écran. Il séparer l'application en trois parties : le modèle, la vue et le contrôleur [4].

Modèle (Model) : En plus de définir la logique métier, le modèle représente les données d'application. Il est en charge du contrôle d'accès, de la gestion et de la manipulation des données. L'interface utilisateur et la manière dont les données sont présentées à l'utilisateur sont inconnues du modèle.

Vue (View) : La Vue représente la partie visible par l'utilisateur de l'application et présente les informations dans le bon format. Les données du modèle sont la seule source d'informations pour la vue, qui manque de toute logique métier.

Contrôleur (Controller) : Le contrôleur sert de conduit entre le modèle et la vue. Il analyse les entrées des utilisateurs et révise le modèle si nécessaire. Le contrôleur reçoit les demandes de l'utilisateur, travaille avec le modèle pour obtenir les informations requises, puis choisit la vue appropriée pour montrer les résultats à l'utilisateur.

2.7 Les prototypes des interfaces de l'application

Pour avoir une vue spécifique sur notre futur interface nous avons créé des prototypage qui est une étape décisive dans la création d'une application. Ce ne seraient pas la version finale des interfaces mais elle nous permettra de définir clairement les fonctions attendues du système.

Ci-dessous, nous présentons les principaux prototypes qui nous semblent importants.

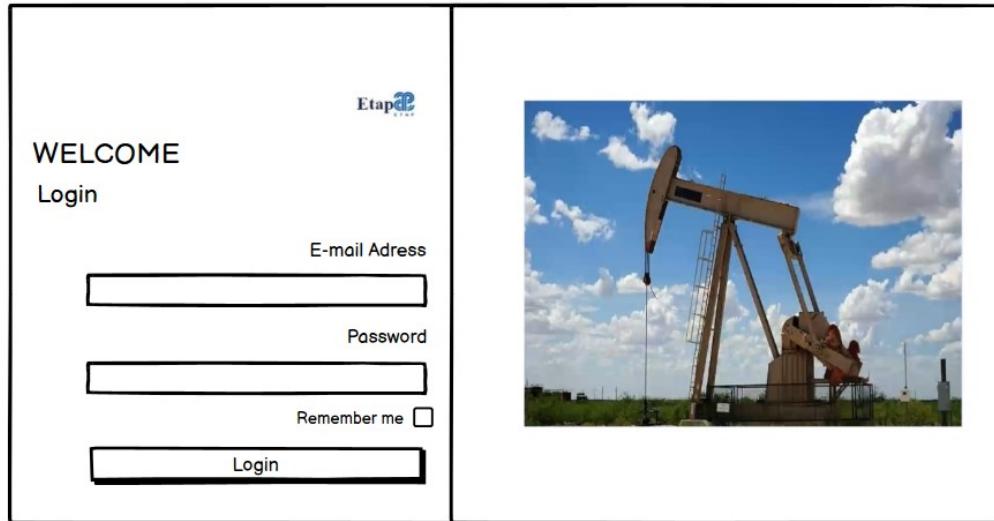


FIGURE 2.8 : Interface « login »

Ce prototype présente l'interface de l'authentification

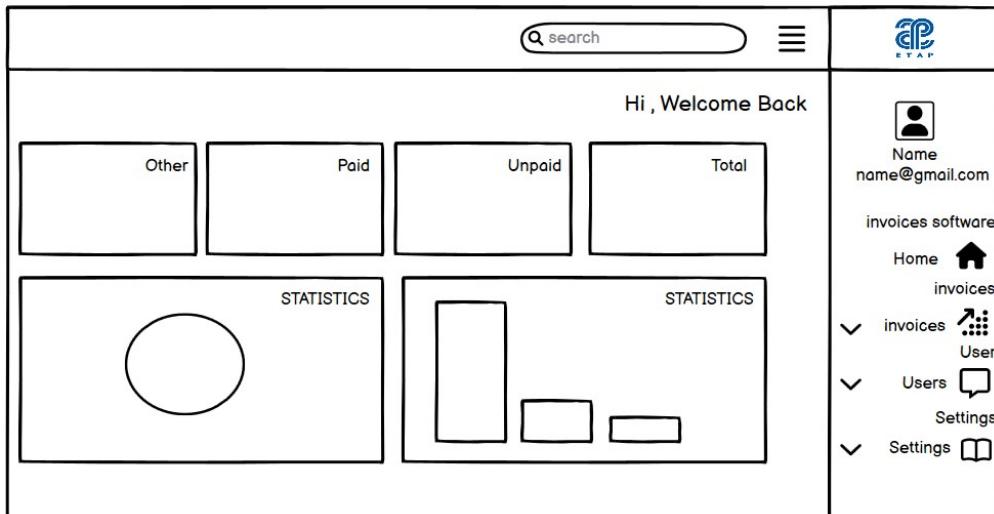


FIGURE 2.9 : Interface « Home page »

Ce prototype présente l'interface du dashboard

The screenshot shows a web-based application interface for managing invoices. At the top right is a user profile icon with the name "name@gmail.com". The main area displays a table titled "INVOICES TABLE" with one row of data. The columns include: Created by (iheb), Note (paid), Status (Total 3770.00), Rate_val (5), Discount (10), Local lab (ETAP), Counter type (gaz), Counter ref (REF-154378), Due_date (2023-07-07), Invoice_date (2023-06-30), invoice_num (#1), and a blank column. Below the table are buttons for Delete, Edit, and Print. At the bottom are navigation buttons for Next, Previous, and a page number indicator (1). A search bar and a "Column visibility" dropdown are at the top left. A sidebar on the right contains links for Home, invoices, INVOICES list, Paid INVOICES, Unpaid INVOICES, INVOICES Other, User, Users, Settings, and Settings.

FIGURE 2.10 : Interface « Gestion factures »

Ce prototype présente l'interface pour gérer les factures

The screenshot shows the "Add invoices\ Invoices" form. It includes fields for: Due date, Invoices date, Invoices number; Local label, Counter type, Counter reference; Total, Discount; VAT, Price; and a large Note field. At the bottom is an "Add" button. The sidebar on the right is identical to Figure 2.10, showing the same navigation links.

FIGURE 2.11 : Interface « Ajouter facture »

Ce prototype présente l'interface pour l'ajout des factures



FIGURE 2.12 : Interface « Imprimer facture »

Ce prototype présente l'interface pour imprimer la facture

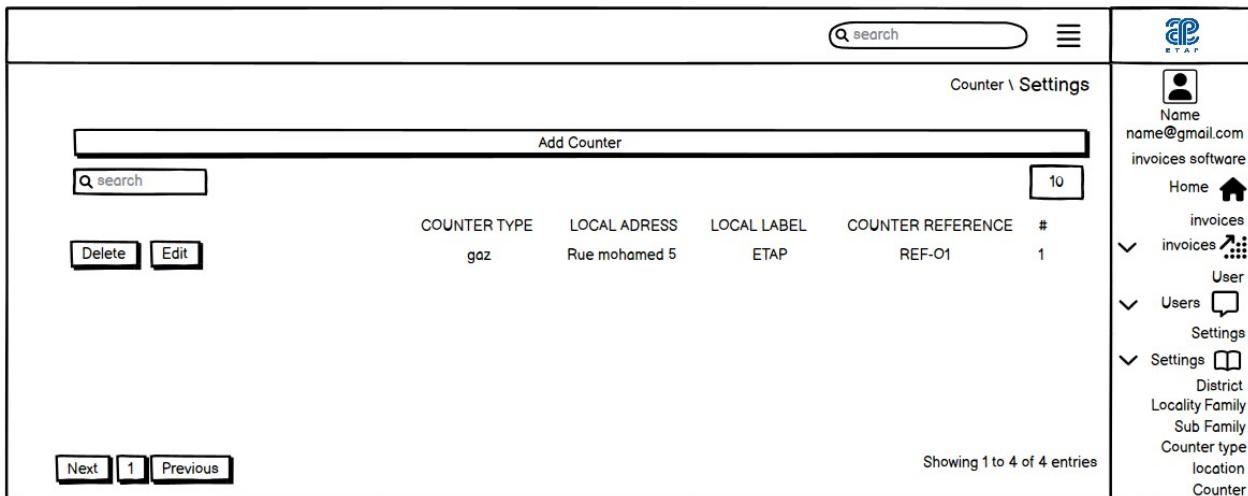


FIGURE 2.13 : Interface « Gestions des compteurs»

Ce prototype présente l'interface pour la gestion des compteurs

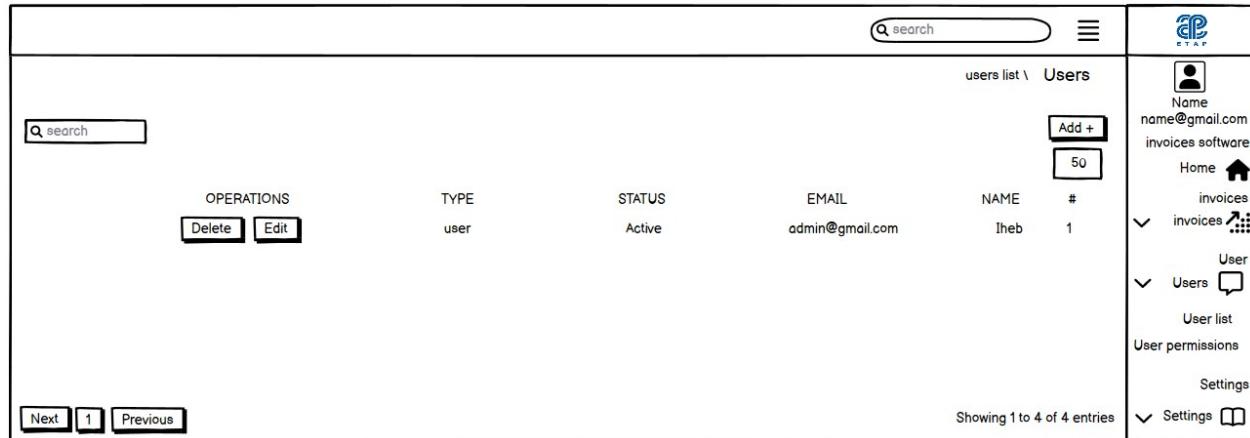


FIGURE 2.14 : Interface « Gestions des utilisateurs »

Ce prototype présente l'interface pour la gestion des utilisateurs

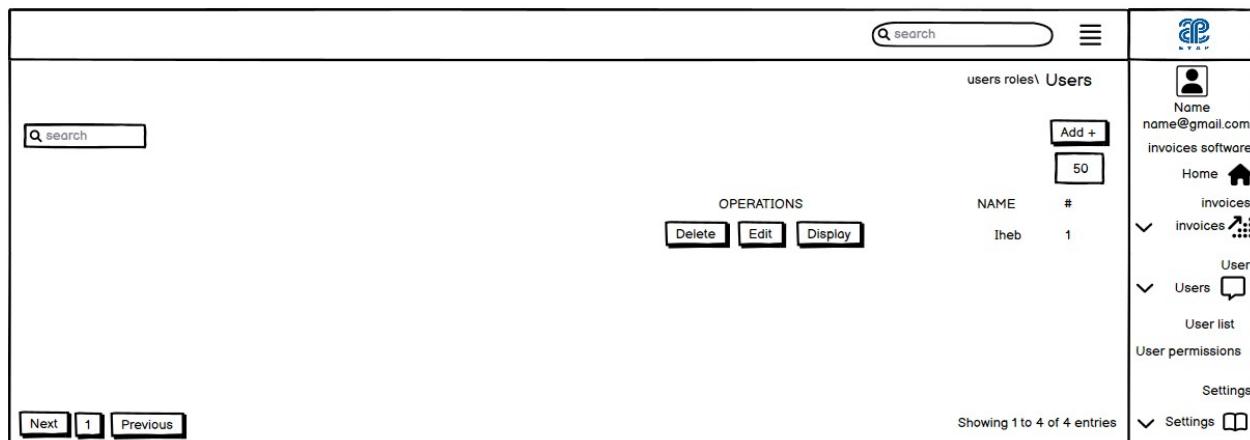


FIGURE 2.15 : Interface « Gestions des rôles »

Ce prototype présente l'interface pour la gestion des rôles

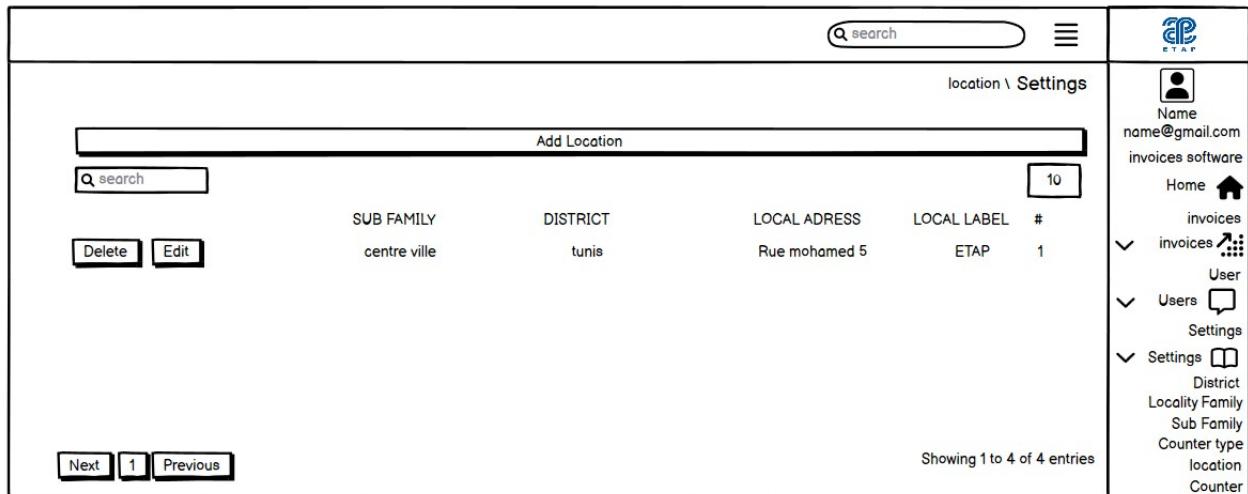


FIGURE 2.16 : Interface « Gestions des locales »

Ce prototype présente l'interface pour la gestion des locales

Conclusion

Nous avons expliqué au sein de ce chapitre les besoins et les spécifications des acteurs, les diagrammes de cas d'utilisations et le diagramme de class. Nous avons montré les caractéristiques du modèle MVC ensuite et les prototypes afin de mieux comprendre le projet.

CONCEPTION

Plan

1	Diagrammes de séquence objets	23
2	Diagramme de classe de conception	26

Introduction

Après la spécification des besoins de notre application nous pouvons commencer la réalisation de l'aspect technique de notre application. Dans ce cas, nous allons présenter dans ce chapitre les différents diagrammes de séquences.

3.1 Diagrammes de séquence objets

3.1.1 Diagrammes de séquence objets « Authentification »

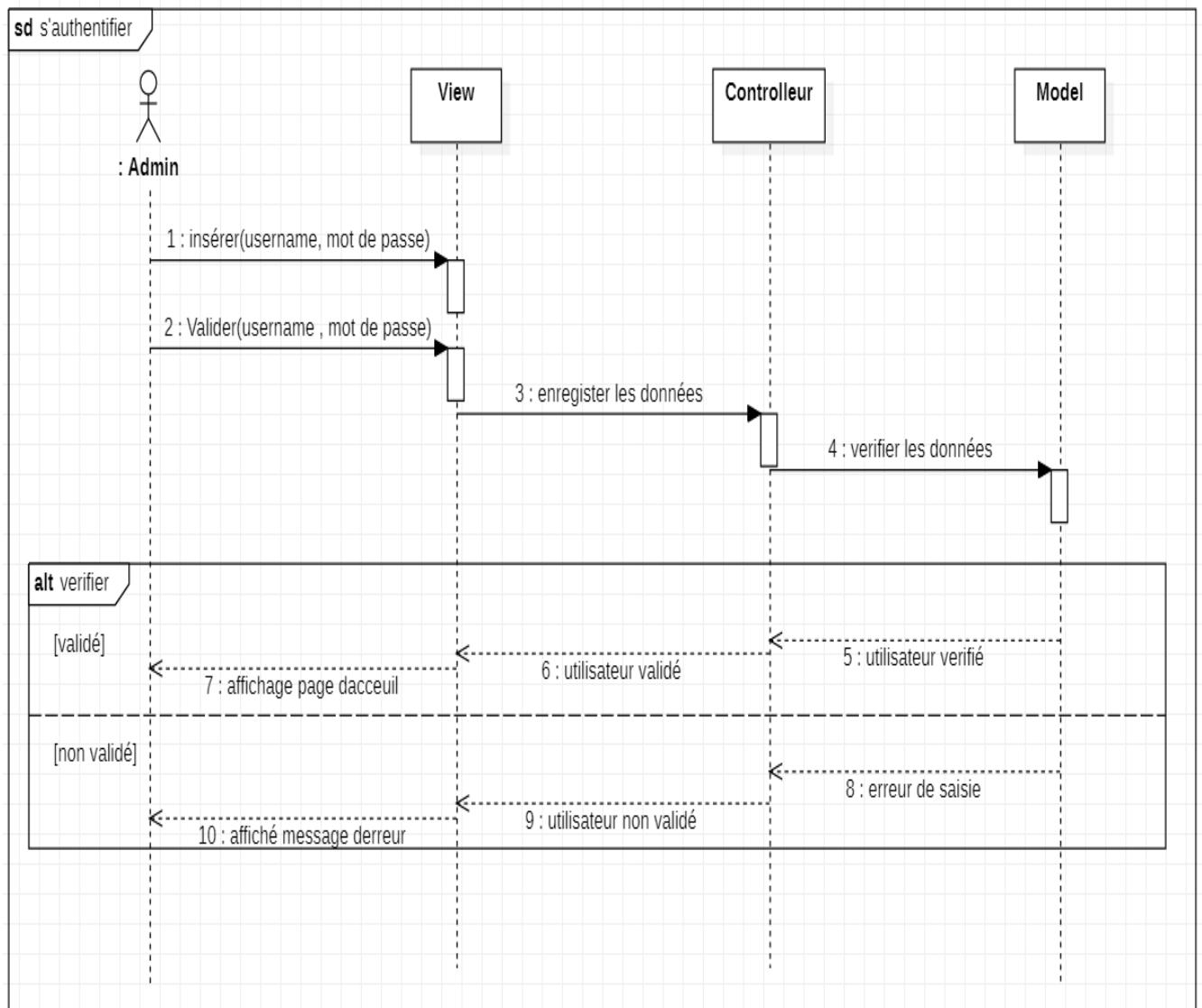


FIGURE 3.1 : Diagramme de cas d'utilisation «Admin - Authentification»

3.1.2 Diagrammes de séquence objets « Gestion facture »

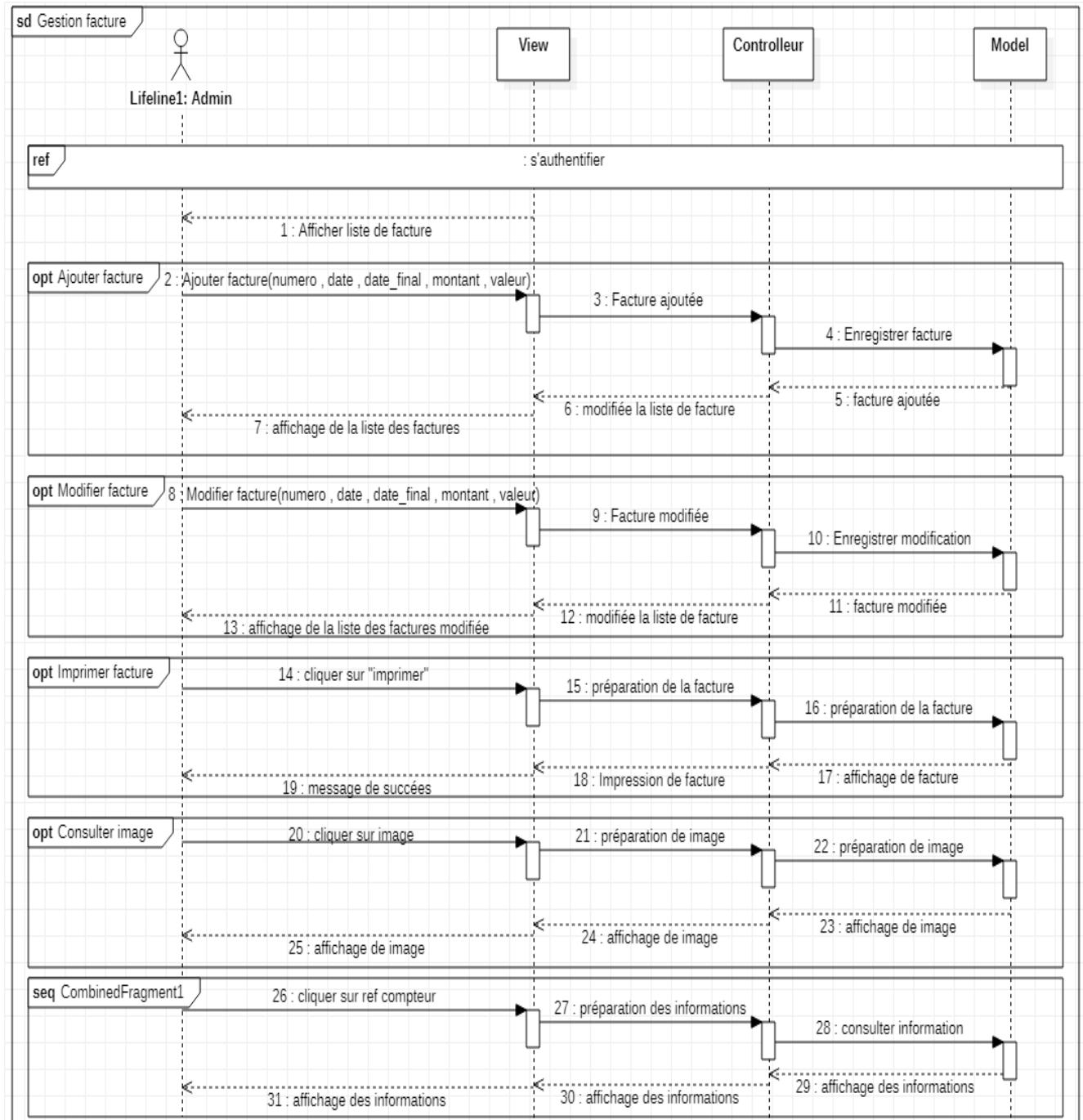


FIGURE 3.2 : Diagramme de cas d'utilisation «Admin - Gestion facture»

3.1.3 Diagrammes de séquence objets « Gestion utilisateur »

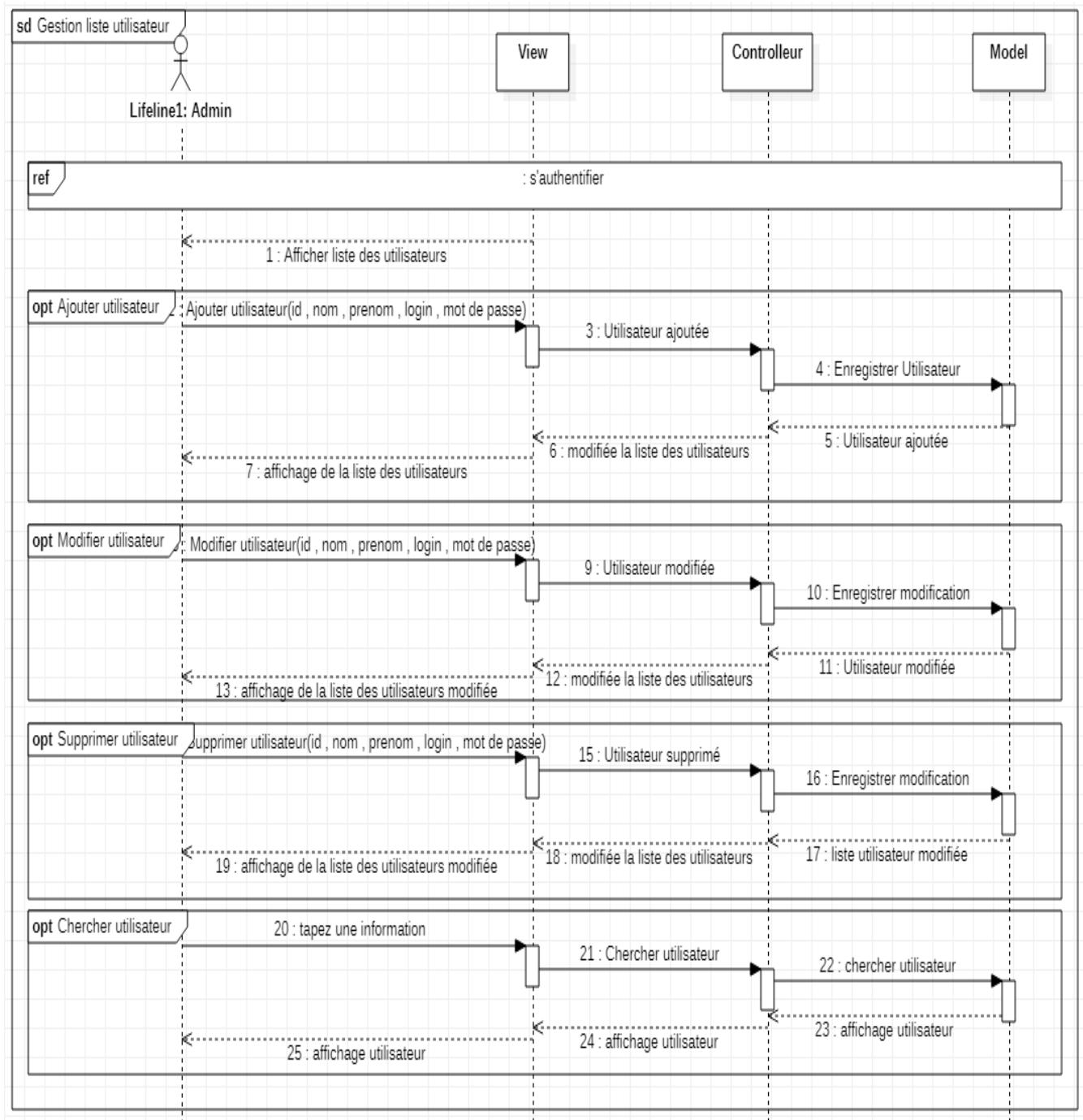


FIGURE 3.3 : Diagramme de cas d'utilisation «Admin - Gestion utilisateur»

3.2 Diagramme de classe de conception

Le diagramme de classes de conception permet de définir la vue de conception statique d'un système et d'en faire ressortir toutes les noms d'attributs et d'opérations.

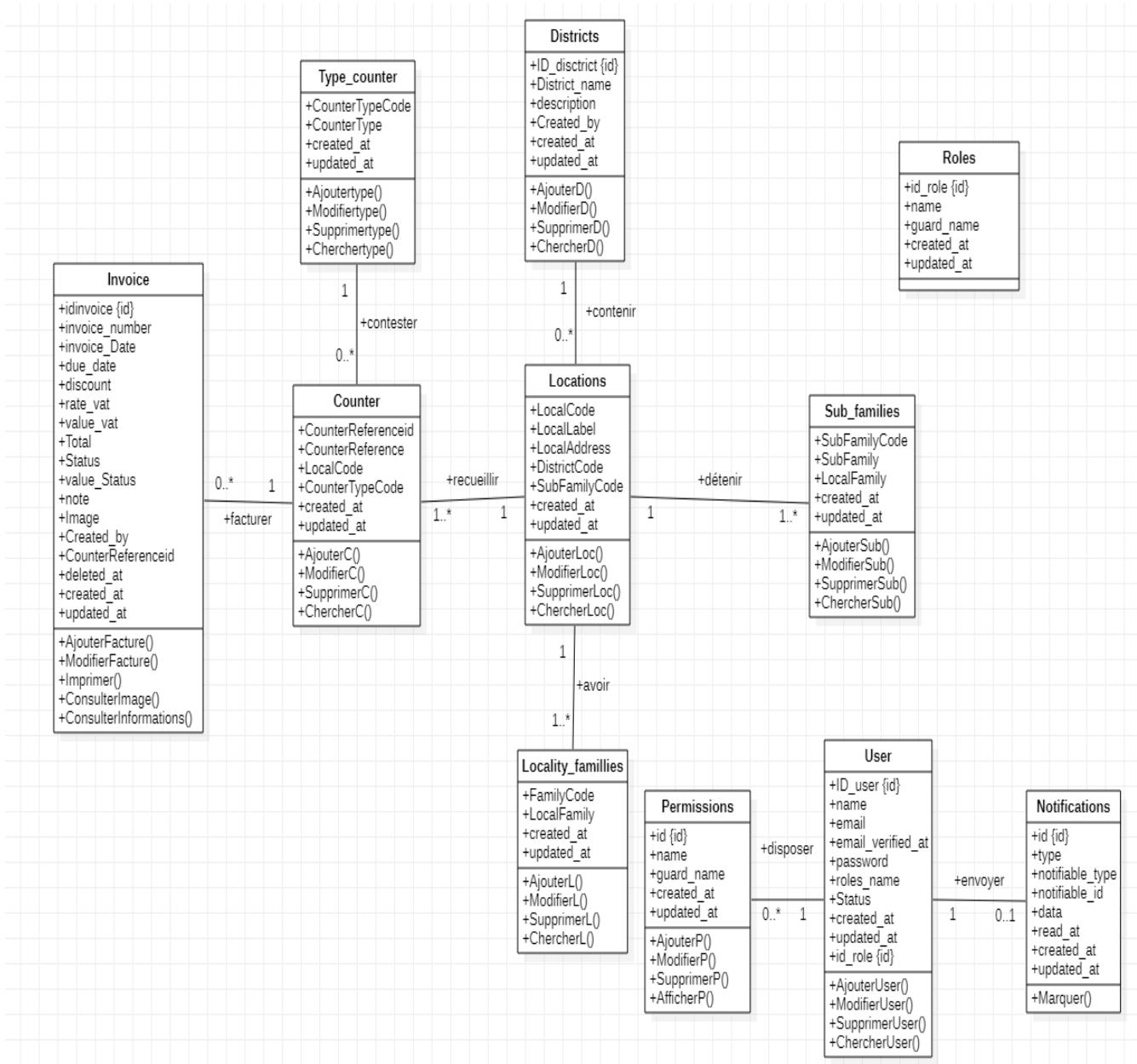


FIGURE 3.4 : Diagramme de classe de conception

Conclusion

Dans ce chapitre on a mis en valeurs les diagrammes de séquences des différentes scénario et le diagramme de classe de conception .

Chapitre 4

RÉALISATION

Plan

1	Outils de réalisation	28
2	Réalisation	30

Introduction

Une fois notre conception établie, nous nous concentrerons maintenant sur l'élaboration de notre application.

4.1 Outils de réalisation



FIGURE 4.1 : visual paradigm

Staruml est **un logiciel de modélisation** qui vas nous aider à faire la conception [5].



FIGURE 4.2 : Laravel

Laravel est un **framework web open-source** écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet[6].



FIGURE 4.3 : XAMPP

XAMPP est un **ensemble de logiciels** permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation,[7].

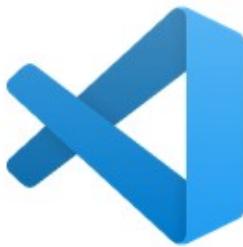


FIGURE 4.4 : Visual Studio Code

Visual studio code est **un éditeur de code** qui n'as pas de relation avec le langage qu'il prend en charge [8].



FIGURE 4.5 : PhpMyAdmin

phpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL[9].



FIGURE 4.6 : GitHub

GitHub est une plate-forme qui permet de travailler ensemble sur des projets en équipe dans plusieurs places [10].

4.2 Réalisation

4.2.1 login

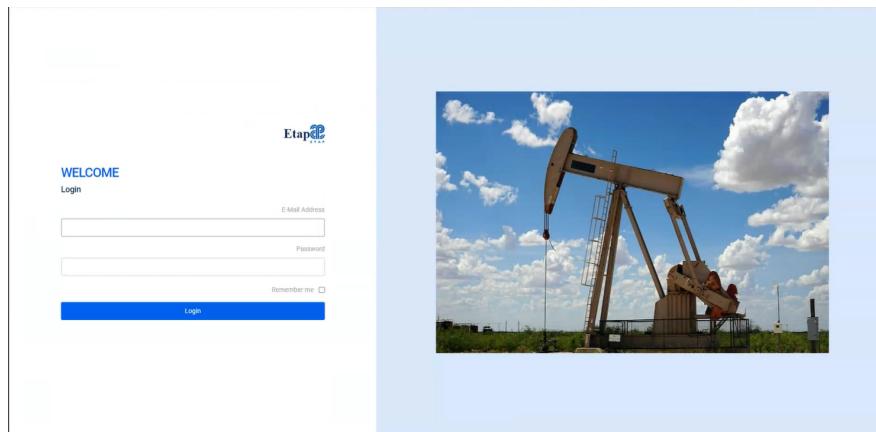


FIGURE 4.7 : Réalisation de la page login

4.2.2 Page initial

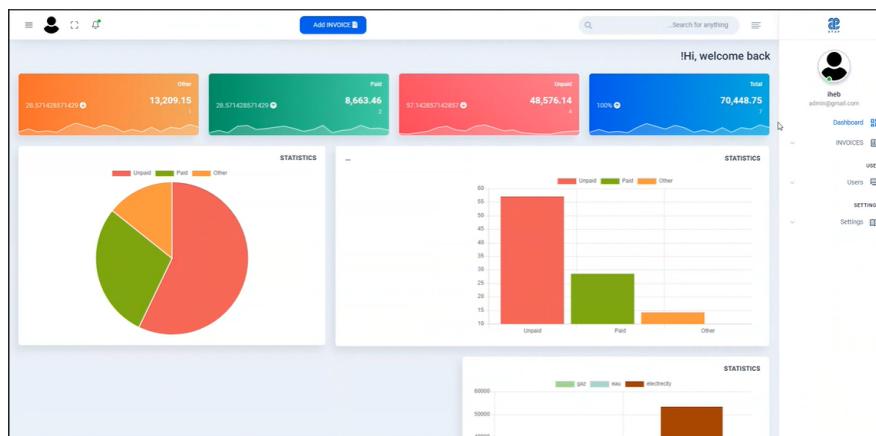


FIGURE 4.8 : Réalisation de la page initial

4.2.3 Gestion région

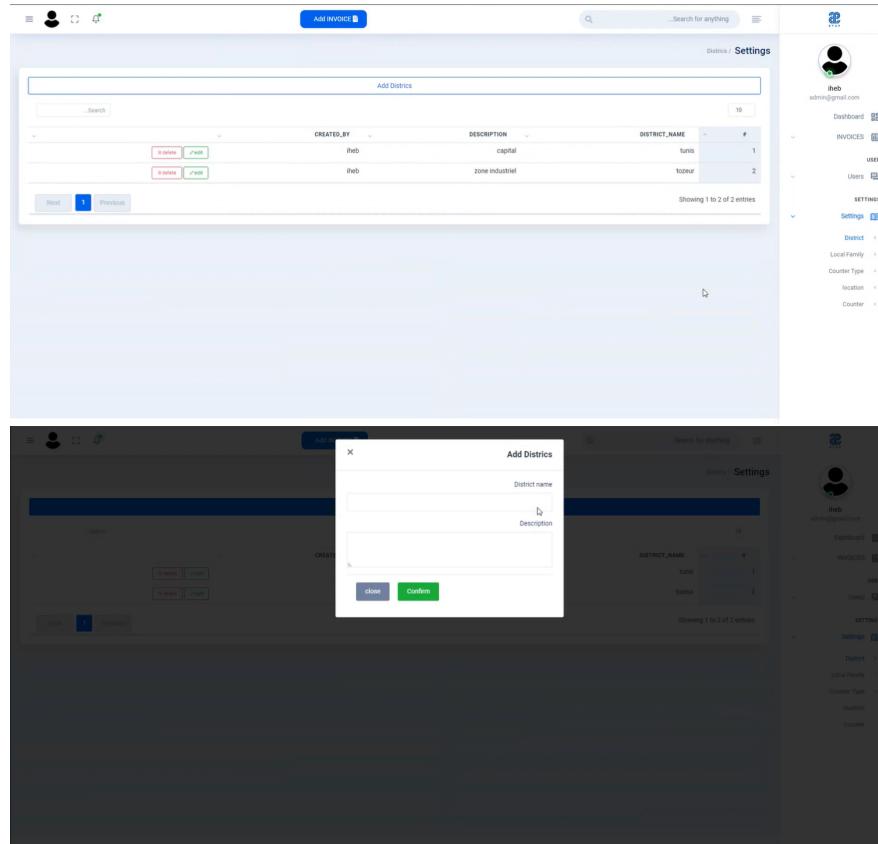


FIGURE 4.9 : Réalisation « Gestion région »

Cette interface nous montre la gestion des régions et l'interface d'ajout d'une région. On peut ajouter, modifier et supprimer comme indiqué dans la **figure 4.9**.

4.2.4 Gestion locale

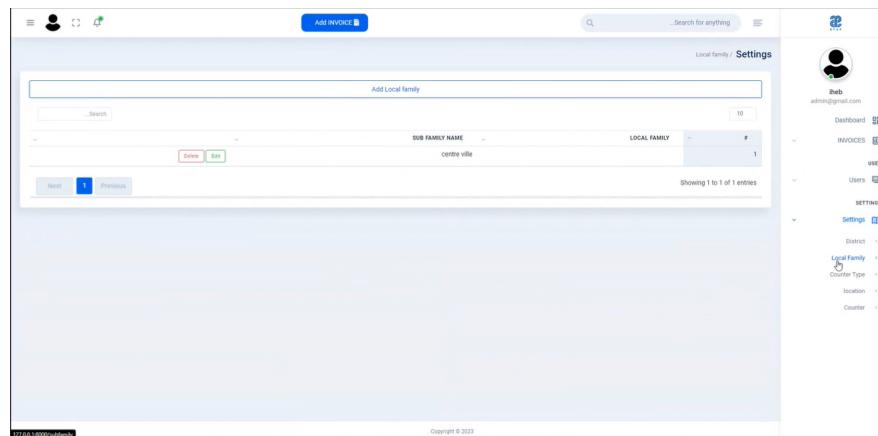


FIGURE 4.10 : Réalisation « Gestion locale »

Cette interface nous montre la gestion des locales. on peut ajouter et supprimer comme indiqué dans la **figure 4.10**.

4.2.5 Gestion emplacements

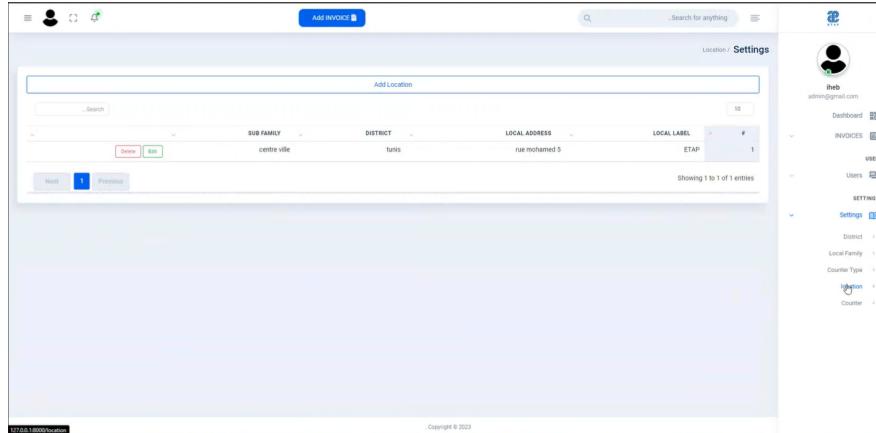


FIGURE 4.11 : Réalisation « Gestion emplacements »

Cette interface nous montre la gestion des emplacements. on peut ajouter, modifier et supprimer comme indiqué dans la **figure 4.11**.

4.2.6 Gestion utilisateurs

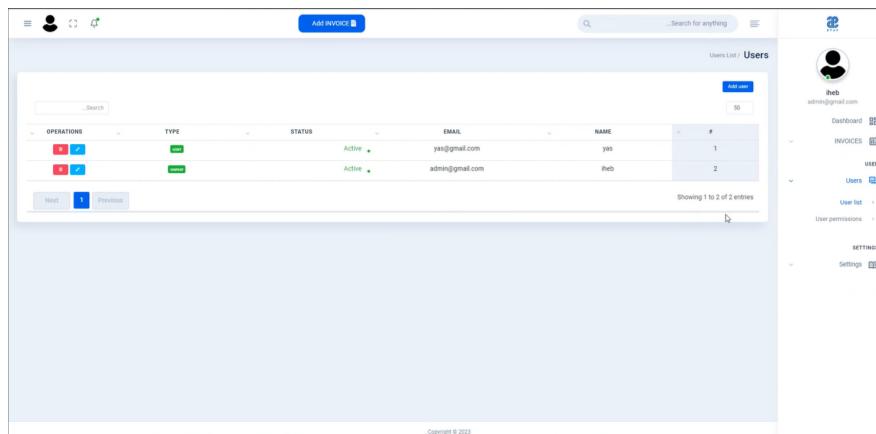


FIGURE 4.12 : Réalisation « Gestion cours »

Cette interface nous montre la gestion des utilisateurs. on peut ajouter, modifier et supprimer comme indiqué dans la **figure 4.12**.

4.2.7 Gestion compteurs

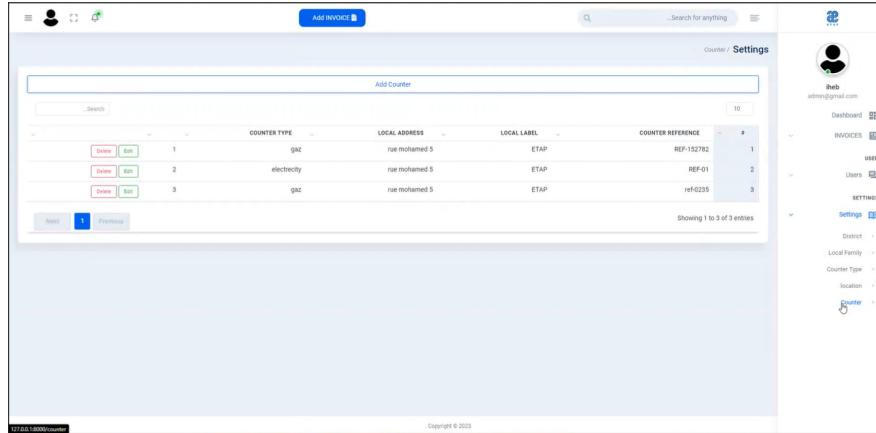


FIGURE 4.13 : Réalisation « Gestion compteurs »

Cette interface nous montre la gestion des compteurs. on peut ajouter, modifier et supprimer comme indiqué dans la **figure 4.13**.

4.2.8 Gestion types de compteurs

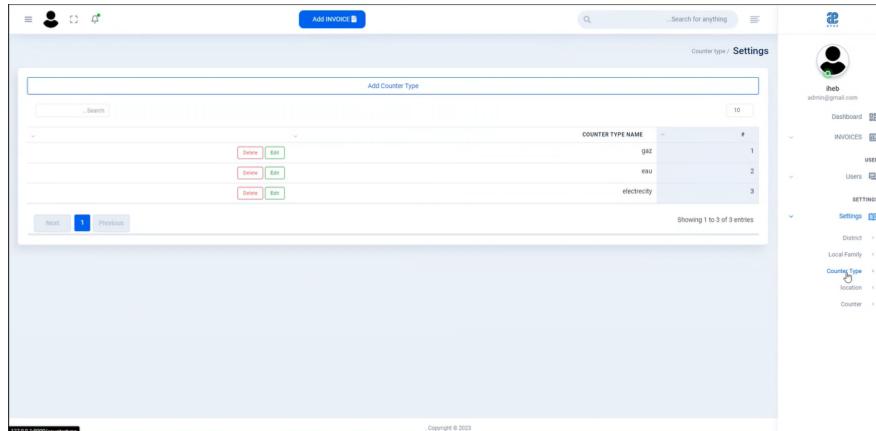
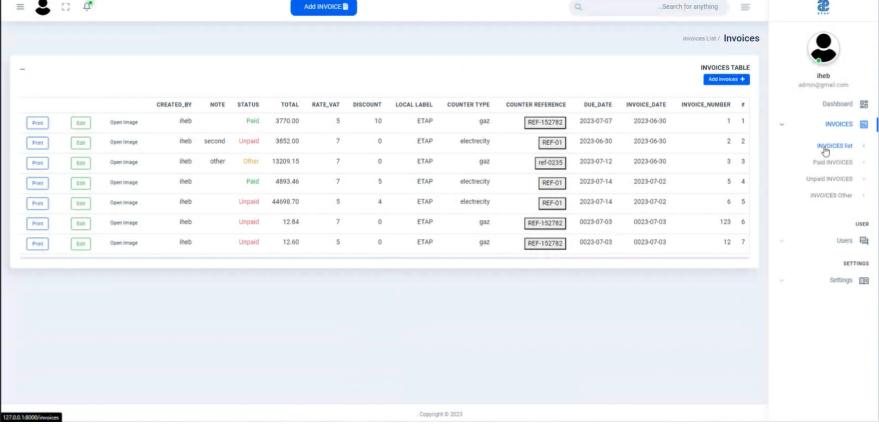


FIGURE 4.14 : Réalisation « Gestion types de compteurs »

Cette interface nous montre la gestion des types de compteurs. on peut ajouter, modifier et supprimer comme indiqué dans la **figure 4.14**.

4.2.9 Interface facture



The screenshot shows a web application for managing invoices. The main area displays a table of invoices with columns including: CREATED_BY, NOTE, STATUS, TOTAL, RATE_VAT, DISCOUNT, LOCAL_LABEL, COUNTER_TYPE, COUNTER_REFERENCE, DUE_DATE, INVOICE_DATE, INVOICE_NUMBER, and #. The table contains 8 rows of data. The sidebar on the right provides navigation links for the user (ihb) and sections such as Dashboard, INVOICES, and SETTINGS.

CREATED_BY	NOTE	STATUS	TOTAL	RATE_VAT	DISCOUNT	LOCAL_LABEL	COUNTER_TYPE	COUNTER_REFERENCE	DUEDATE	INVOICE_DATE	INVOICE_NUMBER	#
[Redacted]	[Redacted]	Paid	3770.00	5	10	ETAP	gaz	REF-152782	2023-07-07	2023-06-30	1	1
[Redacted]	[Redacted]	Open	3852.00	7	0	ETAP	electricity	REF-01	2023-06-30	2023-06-30	2	2
[Redacted]	[Redacted]	Open	13209.15	7	0	ETAP	gaz	ref-0235	2023-07-12	2023-06-30	3	3
[Redacted]	[Redacted]	Open	4893.46	7	5	ETAP	electricity	REF-01	2023-07-14	2023-07-02	5	4
[Redacted]	[Redacted]	Open	44698.70	5	4	ETAP	electricity	REF-01	2023-07-14	2023-07-02	6	5
[Redacted]	[Redacted]	Open	12.84	7	0	ETAP	gaz	REF-152782	2023-07-03	2023-07-03	123	6
[Redacted]	[Redacted]	Open	12.60	5	0	ETAP	gaz	REF-152782	2023-07-03	2023-07-03	12	7

FIGURE 4.15 : Réalisation « Gestion facture »

Cette interface permet à l'utilisateur de gérer ses factures comme indiqué dans la figure 4.15.

Conclusion

La fin de ce chapitre marque donc la fin de notre étude du projet. Nous avons invoqué les diverses interfaces qui illustrent et représentent les principaux modules que nous avons réalisés tout au long de notre projet.

Conclusion générale

En résumé, les entreprises qui souhaitent réduire leurs coûts énergétiques et promouvoir la responsabilité des ressources doivent porter une attention particulière à la précision avec laquelle elles gèrent leur consommation d'électricité, de gaz et d'eau. Dans ces circonstances, ETAP a compris l'importance d'une méthode efficace pour enregistrer les relevés des compteurs, surveiller la consommation et gérer les factures associées. Par conséquent, il s'est avéré crucial de créer une application Web spécifiquement pour atteindre cet objectif.

L'application proposée permettra aux utilisateurs d'accéder simplement à toutes les données pertinentes concernant leur consommation d'électricité, de gaz et d'eau, ce qui simplifiera la gestion des factures. Il disposera de fonctionnalités permettant la saisie de relevés de compteurs, un suivi précis de la consommation et la création de factures personnalisées.

Le rapport décrit dans cette introduction passera en revue les différentes phases de développement de cette application. Il sera divisé en différents chapitres qui permettront aux lecteurs de mieux comprendre l'historique du projet, ses exigences fonctionnelles et non fonctionnelles et ses exigences de conception.

En conclusion, ce rapport donnera un aperçu du projet de développement d'une application de gestion de la consommation d'énergie pour la société ETAP. Il attirera l'attention sur les efforts déployés pour offrir une solution viable et adaptée, permettant aux utilisateurs de gérer leurs ressources énergétiques de manière raisonnable et abordable.

Bibliographie

- [1] <http://www.etap.com.tn/>
- [2] <https://pi.tn/>
- [3] <https://www.ionos.co.uk/digitalguide/websites/web-development/waterfall-methodology/>
- [4] [//wisdomplexus.com/blogs/mvc-vs-microservices/](http://wisdomplexus.com/blogs/mvc-vs-microservices/)
- [5] <https://staruml.io/>
- [6] <https://laravel.com/>
- [7] <https://www.apachefriends.org/fr/index.html>
- [8] <https://code.visualstudio.com/>
- [9] <https://www.phpmyadmin.net/>
- [10] <https://github.blog/>

