

UNIVERSITÉ DE TUNIS EL MANAR

FACULTÉ DES SCIENCES DE TUNIS



RAPPORT DE STAGE D'ÉTÉ

Sujet :

Conception et développement d'une application
de gestion des Factures

Réalisé au sein de L'Entreprise Tunisienne d'Activités Pétrolières



Élaboré par :

SATOURI RANIM

Encadré par :

MANSOURI AMAL

Remerciement

Tout d'abord, je remercie Dieu de m'avoir accordé la santé et la patience pour mener à bien ce projet.

Je souhaite également adresser mes sincères remerciements à l'Entreprise Tunisienne de l'Activité Pétrolière (ETAP) pour m'avoir offert l'opportunité de réaliser ce projet au sein de son organisation. Travailler au sein de l'ETAP a été une expérience enrichissante qui m'a permis de mettre en pratique mes connaissances académiques dans un environnement professionnel stimulant.

Je tiens particulièrement à remercier chaleureusement Mme Mansouri Amal, mon encadrante, pour son soutien inestimable, sa passion pour l'enseignement et son encouragement constant tout au long de ce projet. Ses précieux conseils et son expertise ont grandement contribué à l'aboutissement de ce travail.

Table des matières

Remerciement	2
Introduction générale	8
1 Cadre Général du projet	9
Introduction	10
1.1 Presentation de l'organisme d'accueil	10
1.1.1 Les objectifs de l'ETAP	10
1.1.2 Les principales activités de l'ETAP	10
1.1.3 Organisation générale de l'ETAP	11
1.1.4 Direction informatique	12
1.2 Étude de l'existant	12
1.3 Critique de l'existant	12
1.4 Solutions proposées	13
1.5 Méthodologie de développement	13
1.5.1 Processus adopté	13
1.5.2 Language de modélisation	14
Conclusion	15
2 Analyse et spécification des besoins	16
Introduction	17
2.1 Identification des acteurs	17
2.2 Identification des besoins	17
2.2.1 Les besoins fonctionnels	17

2.2.2	Les besoins non fonctionnels	18
2.3	Spécification des besoins	18
2.3.1	Diagramme de cas d'utilisation global	18
2.3.2	Raffinement et descriptions textuelles des cas d'utilisation	19
	Conclusion	23
3	Architecture et Conception de la solution	24
	Introduction	25
3.1	Architecture de L'application	25
3.2	Conception	26
3.2.1	Vue dynamique	26
3.3	Vue statique	29
	Conclusion	30
4	Réalisation de L'application	31
	Introduction	32
4.0.1	Environnements du travail	32
4.1	Interfaces graphiques de l'application	34
4.1.1	Interface d'authentification :	34
4.1.2	L'interface d'accueil	35
4.1.3	Les interfaces d'ajout	36
4.1.4	interface Admin pour gérer les Factures	38
4.1.5	Interface admin pour gérer les Employés	40
4.1.6	Interface d'états statistiques	40
4.2	Chronogramme de réalisation du projet	41
	Conclusion	42
	Conclusion générale et perspectives	43

Table des figures

1.1	Organisme de L'ETAP	11
1.2	Représentation de la méthodologie cycle en V	14
2.1	Diagramme de cas d'utilisation global	19
2.2	raffinement du cas d'utilisation « Gérer les Factures »	20
2.3	raffinement du cas d'utilisation « Gérer les Employés »	21
3.1	Architecture Logicielle de l'application	25
3.2	Diagramme de séquence du cas d'utilisation « S'authentifier »	27
3.3	Diagramme de séquence du cas d'utilisation « Ajouter une facture »	28
3.4	Diagramme de Classe	29
4.1	Interface : Authentification	34
4.2	Interface : Accueil de l'administrateur	35
4.3	Interface : Accueil de l'Employé	35
4.4	Interface : Ajouter un district	36
4.5	Interface : Ajouter un Local	36
4.6	Interface : Ajouter un compteur	37
4.7	Interface : Ajouter une facture	37
4.8	Interface : Ajouter un Employé	38
4.9	Interface : Consulter les Factures	38
4.10	Interface : Confirmer la supression d'une facture	39
4.11	Interface : Modifier une facture	39

4.12 Interface : consulter les Employés	40
4.13 Interface : selectionner l'année	40
4.14 Interface : dashboard	41
4.15 Diagramme de Gantt	42

Liste des tableaux

2.1	Description textuelle du cas d'utilisation « S'authentifier »	20
2.2	Description textuelle du cas d'utilisation « Supprimer un Utilisateur »	21
2.3	Description textuelle du cas d'utilisation « Modifier une Facture »	22
2.4	Description textuelle du cas d'utilisation « Ajouter un compteur »	23
4.1	caractéristiques de l'ordinateur	32
4.2	Planning de réalisation des tâches tout au long de la période de stage	41

Introduction générale

Dans le monde d'aujourd'hui, l'informatique joue un rôle prépondérant dans le fonctionnement des entreprises. Les applications informatiques ont révolutionné la manière dont les tâches sont accomplies, automatisant des processus complexes et permettant une gestion plus transparente des ressources.

L'Entreprise Tunisienne de l'Activité Pétrolière (ETAP) ne fait pas exception à cette tendance. En tant qu'acteur clé du secteur de l'énergie en Tunisie, l'ETAP doit gérer de manière efficace et transparente ses opérations, y compris la gestion des factures d'eau, d'électricité et de gaz .

En tant qu'étudiant en génie logiciel à la Faculté des Sciences, j'ai eu l'opportunité de mettre en pratique mes connaissances et compétences en participant à un stage d'initiation au sein de l'Entreprise Tunisienne de l'Activité Pétrolière (ETAP). Ce projet consiste en le développement d'une application de gestion des factures dédiée à l'ETAP. Cette application a été conçue pour répondre spécifiquement aux besoins de l'entreprise et faciliter les tâches de l'administrateur, notamment la gestion, la saisie, le stockage et l'analyse des factures d'eau, d'électricité et de gaz. Elle permet également de suivre des états statistiques qui aident à évaluer la consommation et à prendre les décisions nécessaires.

Le présent rapport se focalise sur la description des différentes étapes de la réalisation de notre sujet. En fait, il est organisé en quatre chapitres :

- Le premier chapitre : « Cadre général du projet » englobe la présentation de l'entreprise ETAP, la problématique, la solution proposée et enfin la méthodologie de développement.
- Le deuxième chapitre : « Analyse et spécification des besoins » souligne les acteurs et les besoins fonctionnels et non fonctionnels de notre projet.
- Le troisième chapitre : « Architecture et conception de la solution » décrit l'étude conceptuelle approfondie de l'application.
- Finalement : « Réalisation de l'application » dans lequel nous abordons la phase de développement en expliquant les différentes interfaces de l'application, finissant par le chronogramme de travail.

Chapitre 1

Cadre Général du projet

Plan du chapitre

Introduction

1. Présentation de l'organisme d'accueil
2. Étude de l'existant
3. Critique de l'existant
4. Solutions proposées
5. Méthodologie de développement

Conclusion

Introduction

Ce premier chapitre vise à situer le présent projet dans son contexte général. En premier lieu, nous allons donner une présentation détaillée de l'organisme d'accueil. Ensuite, nous allons étudier et critiquer l'existant. Puis, nous allons énoncer la solution proposée et clôturer par la méthodologie de travail adoptée.

1.1 Présentation de l'organisme d'accueil

L'entreprise Tunisienne de l'Activité Pétrolière (ETAP) est une société industrielle et commerciale publique créée en vertu de la loi numero 72-22 du 10 mars 1972, chargée de la gestion des activités pétrolières et d'exploration et de la production de gaz naturel.

Elle est classée parmi les entreprises publiques à caractère non administratif (EPNA).

1.1.1 Les objectifs de l'ETAP

- Reconstruire la réserve nationale d'hydrocarbures.
- Fournir des produits pour le marché pétrolier domestique à des conditions de coûts favorables.
- Assurer la formation et le développement des cadres tunisiens dans divers secteurs de l'industrie pétrolière.
- Effectuer toutes les recherches liées au pétrole.
- Il est possible d'intervenir dans toutes les industries, du commerce, de la finance, du meuble et de l'immobilier directement ou indirectement liées aux hydrocarbures.
- Assurer la sécurité de l'approvisionnement en hydrocarbures du pays.

1.1.2 Les principales activités de l'ETAP

- Faites attention à toutes les recherches sur la nature du pétrole.
- Vendre du pétrole brut et fournir du gaz naturel au marché local
- Exportez du pétrole brut local vers ETAP.
- Développer les ressources humaines dans le secteur pétrolier.

1.1.3 Organisation générale de l'ETAP

L'ETAP est dirigée par un Président Directeur Général qui représente L'ETAP dans tous les actes civils et administratifs, secondé par un directeur général adjoint et des conseillers. Elle est subdivisée en diverses sections dont cinq directions centrales. Comme le montre la figure ci-dessous, Mon stage est réalisé à la direction centrale des ressources plus précisément à la direction informatique.

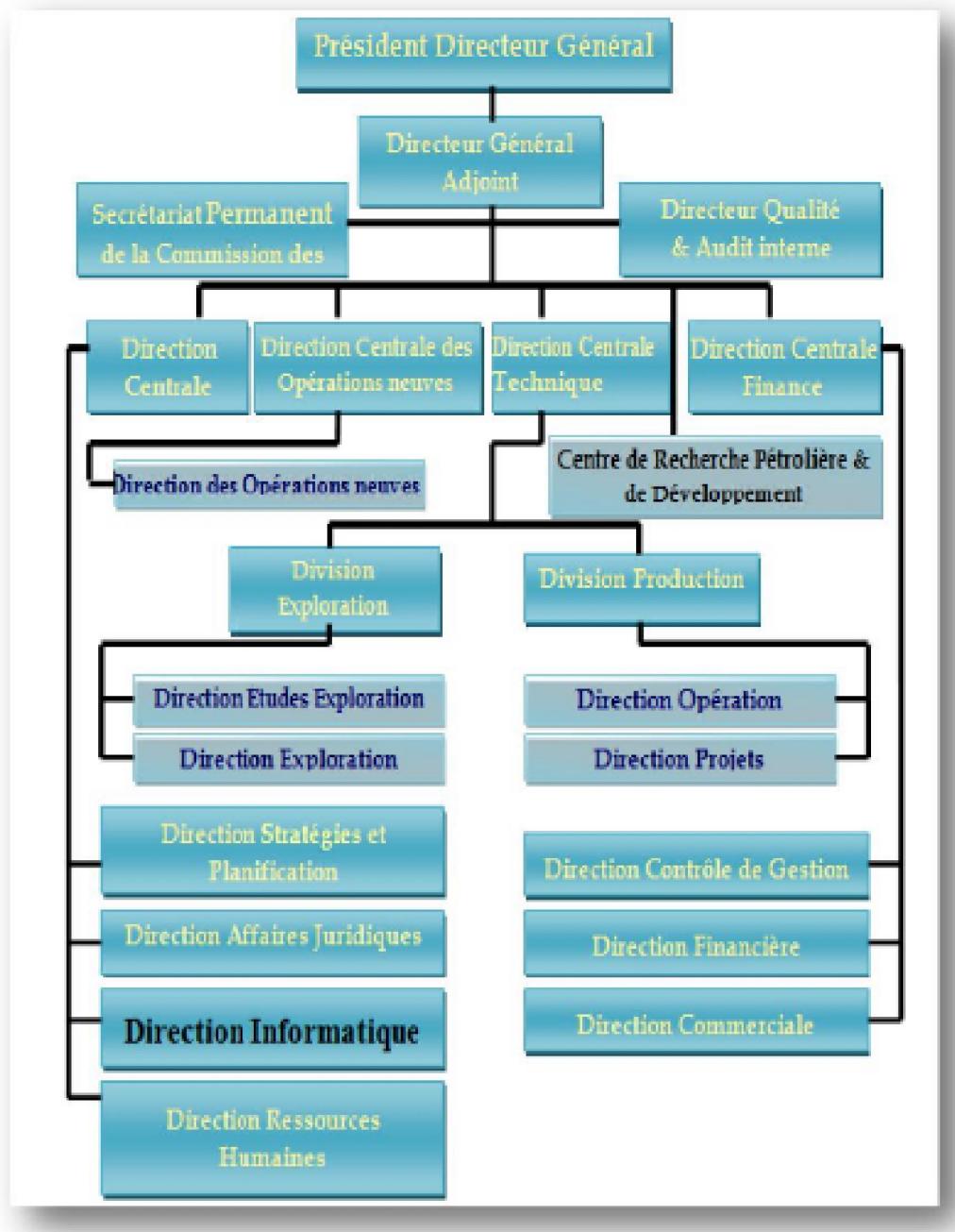


FIGURE 1.1 – Organisme de L'ETAP

1.1.4 Direction informatique

L'ETAP a introduit l'outil informatique dès les années 80 dans le but principal de gérer le patrimoine du pays en bandes sismiques. Le système informatique a également été exploité pour la gestion des applications relatives aux systèmes d'information ETAP (administrative, comptable, gestion, application technique, financière et budgétaires ...) Les principaux projets réalisés au sein de cette direction portent sur :

- Système d'information
- Sécurité informatique
- Réseau et application technique d'exploitation
- Base de données pétrolière

1.2 Étude de l'existant

Afin de comprendre l'application future, il est indispensable de comprendre l'existant et la méthode adoptée pour la gestion des factures. Auparavant, l'ETAP utilisait principalement des feuilles de calcul Excel pour gérer ses factures. Cela signifiait que les données étaient dispersées et souvent redondantes, ce qui compliquait la recherche et l'analyse des informations.

1.3 Critique de l'existant

Après avoir mené une étude approfondie du système actuel, nous avons trouvé diverses critiques :

- Manque de centralisation des données
- Risque de perte de données
- Absence de suivi automatisé

Il est essentiel de souligner que le système existant ne répond pas adéquatement aux besoins en matière de gestion des factures de l'entreprise. Afin de remédier à ces lacunes et de répondre aux besoins croissants de l'entreprise, un nouveau système de gestion des factures est nécessaire.

1.4 Solutions proposées

Face aux limitations du système existant, il est impératif de mettre en place une solution innovante et efficace pour la gestion des factures au sein de l'entreprise. La solution proposée consiste en la conception et la mise en œuvre d'une application de gestion des factures. La solution proposée présentera les caractéristiques clés suivantes :

- Interface conviviale : Une interface utilisateur conviviale permettra aux utilisateurs de naviguer facilement dans l'application et d'accéder rapidement aux informations pertinentes.
- Centralisation des données : Toutes les données des factures seront centralisées dans une base de données sécurisée, éliminant ainsi la dispersion des informations.
- Analyse avancée : L'application offrira des outils d'analyse avancés pour évaluer les coûts des factures par année, ainsi que la consommation d'eau, d'électricité et de gaz par local et par an.

1.5 Méthodologie de développement

1.5.1 Processus adopté

Dans un projet informatique, il est indispensable de choisir une bonne démarche pour atteindre les objectifs visés. La réduction du temps, du cout de développement et des risques d'erreurs, sont les soucis majeurs de toutes équipes de développement.

Pour bien conduire notre projet, nous avons opté pour le Cycle de vie en V. Notre choix s'est basé sur les exigences de ce projet et sur les atouts de ce cycle de vie.

Le modèle en V est parcouru de gauche à droite suivant la forme de la lettre « V » dont la branche descendante contient toutes les étapes de la conception du projet, et la branche montante toutes les étapes de tests du projet.

Le modèle en V demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé. Il permet d'anticiper sur les phases ultérieures de développement du produit.

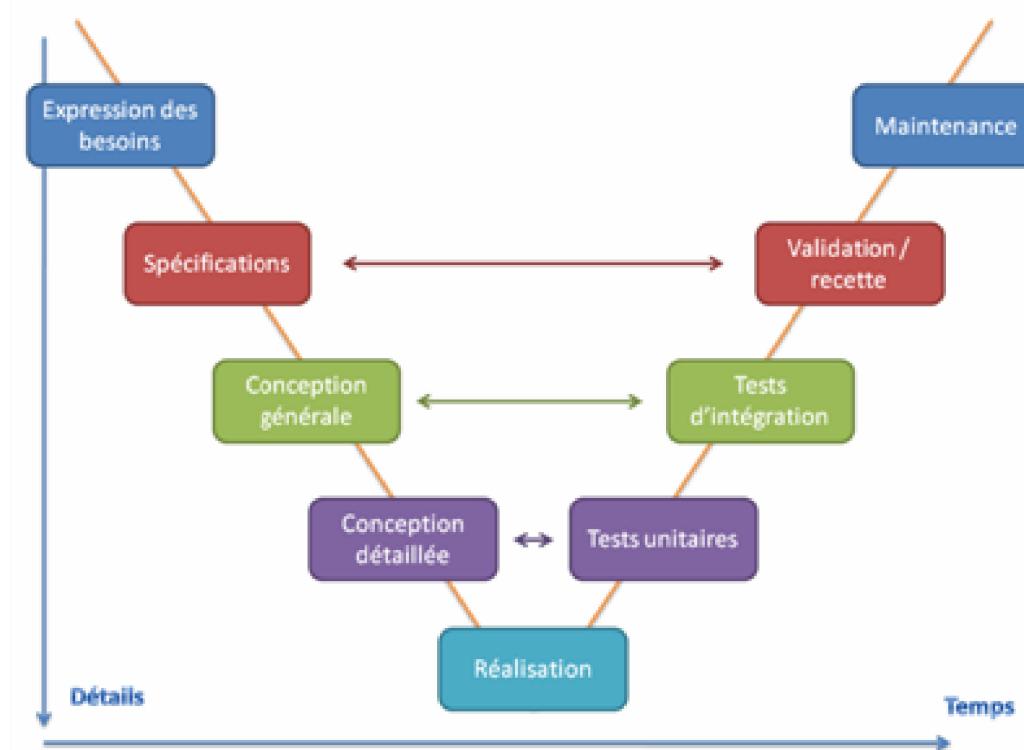


FIGURE 1.2 – Représentation de la méthodologie cycle en V

1.5.2 Language de modélisation

Après le choix de la méthodologie, nous avons besoin d'un langage de modélisation unifiée pour la modélisation de notre projet dans le but de maîtriser sa complexité et d'assurer sa cohérence. Afin de concevoir notre système, nous avons choisi l'UML (Unified Modeling Language) comme un langage de modélisation.

Ce choix est basé sur les points forts de ce langage qui est standard et offre la possibilité de schématiser graphiquement des systèmes complexes sous un format simplifié et normalisé.

L'UML propose divers diagrammes de modélisations, parmi lesquelles, ceux que nous allons utiliser sont :

- ✓ **Le diagramme de cas d'utilisation** qui est un diagramme fonctionnel, il permet de représenter les interactions entre les utilisateurs et le système

- ✓ **Le diagramme de classes** qui est un diagramme statique, il permet de représenter les éléments du système (classes et packages), leur contenu (attributs et méthodes) ainsi que les associations qui les relient.
- ✓ **Le diagramme de séquence** qui est un diagramme dynamique, il permet de représenter temporellement l'échange entre l'utilisateur et le système.

Conclusion

Ce premier chapitre nous a permis de mettre le projet dans son cadre général en présentant l'organisme d'accueil l'ETAP, l'étude et la critique de l'existant ainsi qu'une analyse et description de la méthodologie de développement qui seront utilisés dans ce projet.

Maintenant nous sommes en mesure d'entamer notre deuxième chapitre qui est l'analyse et la spécification des besoins.

Chapitre 2

Analyse et spécification des besoins

Plan du chapitre

Introduction

1. Identification des acteurs
2. Identification des besoins
3. Diagramme des cas d'utilisation global

Conclusion

Introduction

L'analyse et la spécification des besoins représentent une phase primordiale dans la méthodologie de travail adopté, à savoir le cycle de vie en V. En outre, les exigences définies dans ce chapitre sont les fonctionnalités que nous envisageons à implémenter dans l'application. En premier lieu, nous allons identifier les acteurs du système. Ensuite, nous allons présenter les besoins de ce projet. Puis, nous allons clôturer par spécifier les besoins.

2.1 Identification des acteurs

Un acteur est la personne ou le logiciel qui interagit avec notre système afin de répondre à un besoin bien déterminé.

Dans notre application, nous avons identifié essentiellement deux acteurs à savoir l'Administrateur et l'Employé :

- L'Admin : c'est la personne qui gère les tâches administratives, à savoir gérer les employés ,les factures.
- L'employé : C'est la personne qui a le droit d'ajouter et de consulter les factures

2.2 Identification des besoins

L'identification des besoins consiste à traduire les objectifs du projet en un ensemble de besoins fonctionnels et non fonctionnels.

2.2.1 Les besoins fonctionnels

Dans cette partie, nous présentons les différents besoins fonctionnels que notre système doit répondre.
Administrateur :

- S'authentifier : L'administrateur est capable d'accéder aux fonctionnalités de l'application en se connectant.
- Gérer les Employés : Consulter , Ajouter, supprimer et modifier les informations des Employés.

- Gérer les factures : Consulter , Ajouter, supprimer et modifier les Factures.
- Suivre les états statistiques

L'Employé :

- S'authentifier : L'employé est capable d'accéder aux fonctionnalités de l'application en se connectant.
- Gérer les factures : Consulter et Ajouter les Factures.
- Suivre les états statistiques

2.2.2 Les besoins non fonctionnels

- L'ergonomie : l'application doit fournir des interfaces simples, conviviales et bien structurées afin que l'utilisateur puisse l'exploiter sans aucunes difficultés.
- L'extensibilité : le code doit être clair et le système doit être conforme à une architecture évolutive pour permettre de renouveler ou encore ajouter des nouvelles fonctionnalités.
- Besoins de sécurité : l'application doit être hautement sécurisé ,tout accès doit être protégé par un mot de passe et un privilège d'accès.

2.3 Spécification des besoins

Dans cette partie, nous spécifions en détail les besoins fonctionnels de notre système d'une manière formelle en utilisant le diagramme de cas d'utilisation du langage de modélisation UML et des descriptions textuelles.

2.3.1 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation général nous donne une vision globale sur l'application en identifiant les acteurs et en structurant les besoins fonctionnels relatifs à l'application.

La figure 2.1 illustre le diagramme de cas d'utilisation global de notre application.

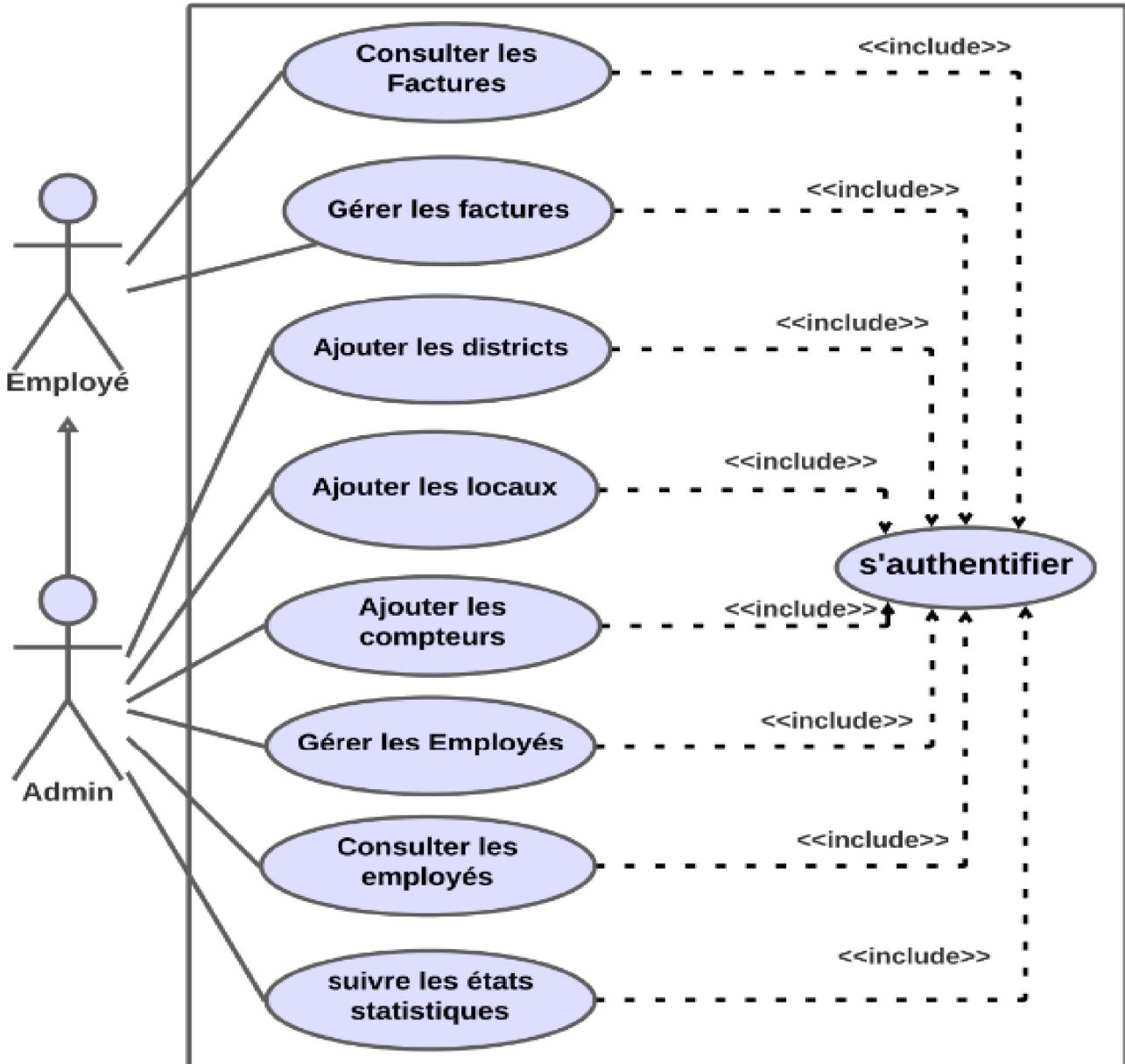


FIGURE 2.1 – Diagramme de cas d'utilisation global

2.3.2 Raffinement et descriptions textuelles des cas d'utilisation

Dans ce qui suit, nous allons raffiner quelques cas d'utilisation qu'on juge prioritaires et les décrire par des descriptions textuelles.

Nous allons commencer par une description textuelle du cas d'utilisation « S'authentifier » représenté par le tableau 2.1.

SOMMAIRE D'IDENTIFICATION	
Titre	S'authentifier
Objectif	Permet aux utilisateurs de s'authentifier sur l'application
Acteurs	Administrateur, Employé
DESCRIPTION DES ENCHAINEMENTS	
précondition :	
<ul style="list-style-type: none"> • l'utilisateur n'est pas authentifié sur l'application 	
Scénario nominal :	
<ol style="list-style-type: none"> 1. Le système affiche le formulaire de connection. 2. L'utilisateur saisit son identifiant et son mot de passe. 3. L'utilisateur clique sur le bouton « Se Connecter ». 4. L'utilisateur accède à la page d'accueil . 	
Scénario alternatif :	
<ul style="list-style-type: none"> • Si l'identifiant et/ou le mot de passe sont manquants ou invalides , le système affiche un message d'erreur. 	
Post condition :	
<ul style="list-style-type: none"> • L'utilisateur est authentifié et redirigé vers la page d'accueil. 	

TABLE 2.1 – Description textuelle du cas d'utilisation « S'authentifier »

Le raffinement du cas d'utilisation « Gérer les Factures » est représenté par la figure 2.2

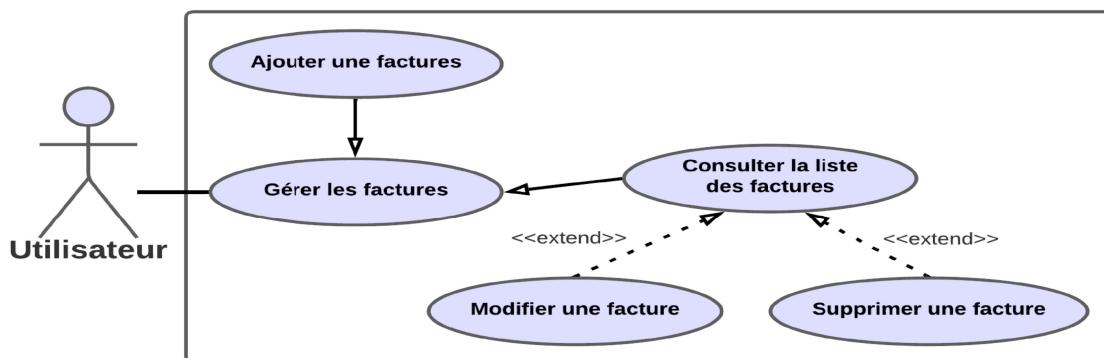


FIGURE 2.2 – raffinement du cas d'utilisation « Gérer les Factures »

Le raffinement du cas d'utilisation « Gérer les Employés » est représenté par la figure 2.3

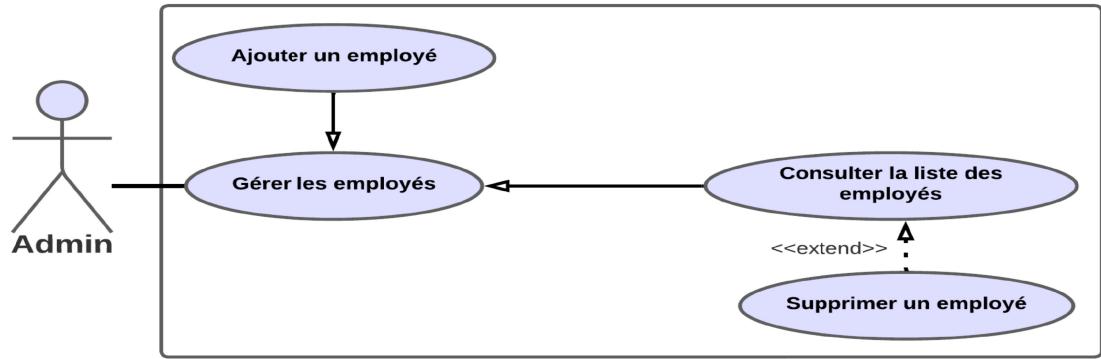


FIGURE 2.3 – raffinement du cas d'utilisation « Gérer les Employés »

SOMMAIRE D'IDENTIFICATION	
Titre	Supprimer un Employé
Objectif	Permet à l'administrateur de Supprimer un Employé de l'application .
Acteurs	Administrateur
DESCRIPTION DES ENCHAINEMENTS	
précondition : <ul style="list-style-type: none"> • L'administrateur est authentifié sur l'application. • Il y a au moins un Employé ajouté. 	
Scénario nominal : <ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton « Consulter les Employé » . 2. Le système retourne la liste d'employés. 3. L'administrateur choisit un Employé et clique sur le bouton « Supprimer » . 4. Le système retourne une boîte de confirmation. 5. L'administrateur confirme la suppression. 6. Le système affiche un message indiquant le succès de l'opération. 	
Scénario alternatif : <ul style="list-style-type: none"> • L'administrateur annule la suppression de l'employé. 	
Post condition : <ul style="list-style-type: none"> • L'employé choisi est supprimé. 	

TABLE 2.2 – Description textuelle du cas d'utilisation « Supprimer un Utilisateur »

SOMMAIRE D'IDENTIFICATION	
Titre	Modifier une Facture
Objectif	Permet à l'utilisateur de modifier une facture.
Acteurs	Administrateur,Employé
DESCRIPTION DES ENCHAINEMENTS	
<p>précondition :</p> <ul style="list-style-type: none"> • L'utilisateur est authentifié sur l'application. • Il y a au moins une facture ajoutée. 	
<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1. L'utilisateur clique sur le bouton « Consulter les Factures » . 2. Le système retourne la liste des factures. 3. L'utilisateur choisit une facture et clique sur le bouton « Modifier » . 4. Le système Retourne un formulaire rempli des données de la facture choisit. 5. L'utilisateur modifie les données . 6. L'utilisateur clique sur le bouton « Modifier ». 7. Le système retourne une boite de confirmation. 8. L'utilisateur confirme la suppression. 9. Le système affiche un message indiquant le succès de l'opération. 	
<p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Le système affiche un message d'erreur si les informations entrées sont manquantes ou erronées. 	
<p>Post condition :</p> <ul style="list-style-type: none"> • La facture est Modifiée. 	

TABLE 2.3 – Description textuelle du cas d'utilisation « Modifier une Facture »

SOMMAIRE D'IDENTIFICATION	
Titre	Ajouter un compteur
Objectif	Permet à l'administrateur d'Ajouter un compteur .
Acteurs	Administrateur
DESCRIPTION DES ENCHAINEMENTS	
précondition : <ul style="list-style-type: none"> • L'administrateur est authentifié sur l'application. • Il y a au moins un District et un Local ajouté. 	
Scénario nominal : <ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton « Ajouter un compteur » . 2. Le système retourne un formulaire . 3. L'administrateur choisit le District et le local a partir du menu déroulant. 4. L'administrateur choisit le type de compteur. 5. L'administrateur saisit la référence du compteur et clique sur le bouton « Ajouter ». 6. Le système affiche un message indiquant le succès de l'opération. 	
Scénario alternatif : <ul style="list-style-type: none"> • Le système affiche un message d'erreur si les informations entrées sont manquantes ou erronées. 	
Post condition : <ul style="list-style-type: none"> • Le compteur est ajouté. 	

TABLE 2.4 – Description textuelle du cas d'utilisation « Ajouter un compteur »

Conclusion

Le travail réalisé dans ce chapitre nous a conduits à construire une meilleure vision de la solution puisque nous avons dégagé les acteurs, identifié les besoins fonctionnels et non fonctionnels et décrit la spécification technique de notre projet.

Le chapitre suivant sera consacré à l'étude conceptuelle de notre application.

Chapitre 3

Architecture et Conception de la solution

Plan du chapitre

Introduction

1. Architecture de L'application
2. Conception

Conclusion

Introduction

La conception est une étape très importante dans la réalisation de notre application, elle permet de concevoir la base de données de notre futur système d'information pour réduire la complexité et lever les ambiguïtés. De plus, elle peut acquérir une meilleure compréhension du système et ainsi gagner du temps. En premier lieu, nous allons décrire l'architecture de notre application. Ensuite, nous allons nous intéresser à la conception du système.

3.1 Architecture de L'application

La conception de l'architecture logicielle représente une étape primordiale dans le processus de conception .Elle dépend de plusieurs facteurs et aide à réaliser les besoins non fonctionnels de l'application comme l'extensibilité et l'évolutivité.

Dans le cadre de ce projet, on a choisi une architecture 1-tier pour développer une application bureau.

Cette architecture est également appelée « client lourd » ou « monolithique ». Dans cette approche, l'intégralité de l'application réside et s'exécute localement sur l'ordinateur de l'utilisateur. L'interface utilisateur, le logique métier et l'accès à la base de données MySQL sont tous inclus dans un code source unique. Cela signifie que notre application bureau est une entité autonome qui gère toutes les fonctionnalités sans dépendre de composants de serveur externes.

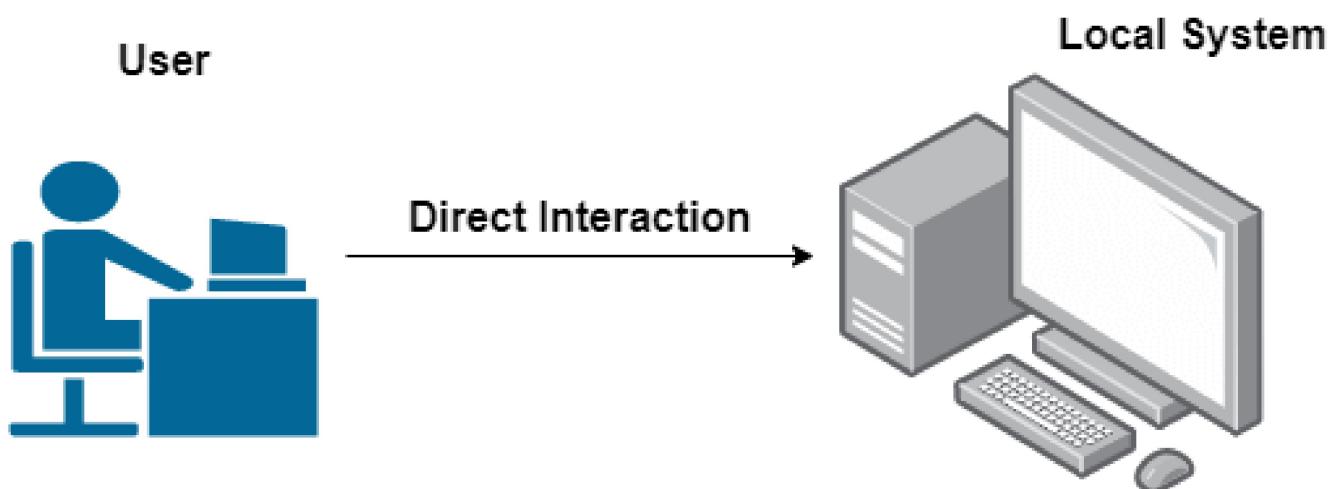


FIGURE 3.1 – Architecture Logicielle de l'application

L'avantage de cette architecture est sa simplicité car elle ne nécessite pas la mise en place de serveurs ou services supplémentaires.

Cependant, il peut présenter des limites en termes de maintenabilité et d'évolutivité à mesure que l'application se développe.

Cette architecture est adaptée aux applications de bureau autonome, où les utilisateurs exécutent l'application localement et n'ont pas besoin d'un accès à distance aux données ou à la logique de l'application.

3.2 Conception

3.2.1 Vue dynamique

Le diagramme de séquence est un diagramme qui décrit l'aspect dynamique du système, il permet de représenter, de façon séquentielle, l'échange entre des acteurs du système et des instances de classes, de composants et de sous-systèmes.

Nous allons présenter ci-après les diagrammes de séquences, que nous avons jugés importants pour la réalisation de notre application.

3.2.1.1 Diagramme de séquence « Authentification »

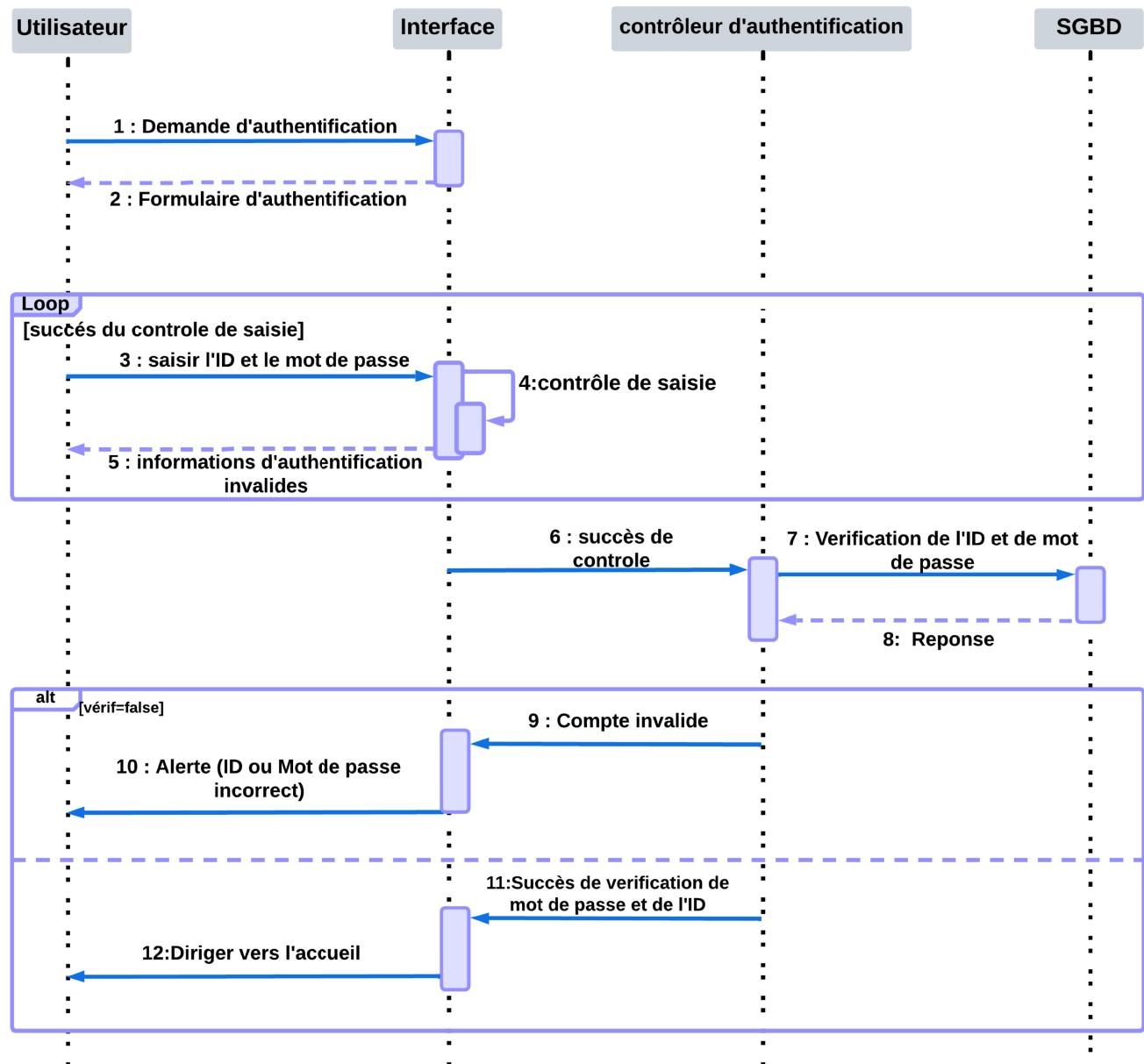


FIGURE 3.2 – Diagramme de séquence du cas d'utilisation « S'authentifier »

Pour s'authentifier, un administrateur/Employé doit saisir son identifiant et son mot de passe, si les données saisies sont correctes alors une session sera ouverte pour lui et il sera redirigé à la page d'accueil de l'application appropriée pour son espace. Si les données sont invalides alors un message d'erreur sera affiché demandant à l'utilisateur de saisir de nouveau l'identifiant et le mot de passe valides.

3.2.1.2 Diagramme de séquence « Ajouter une facture »

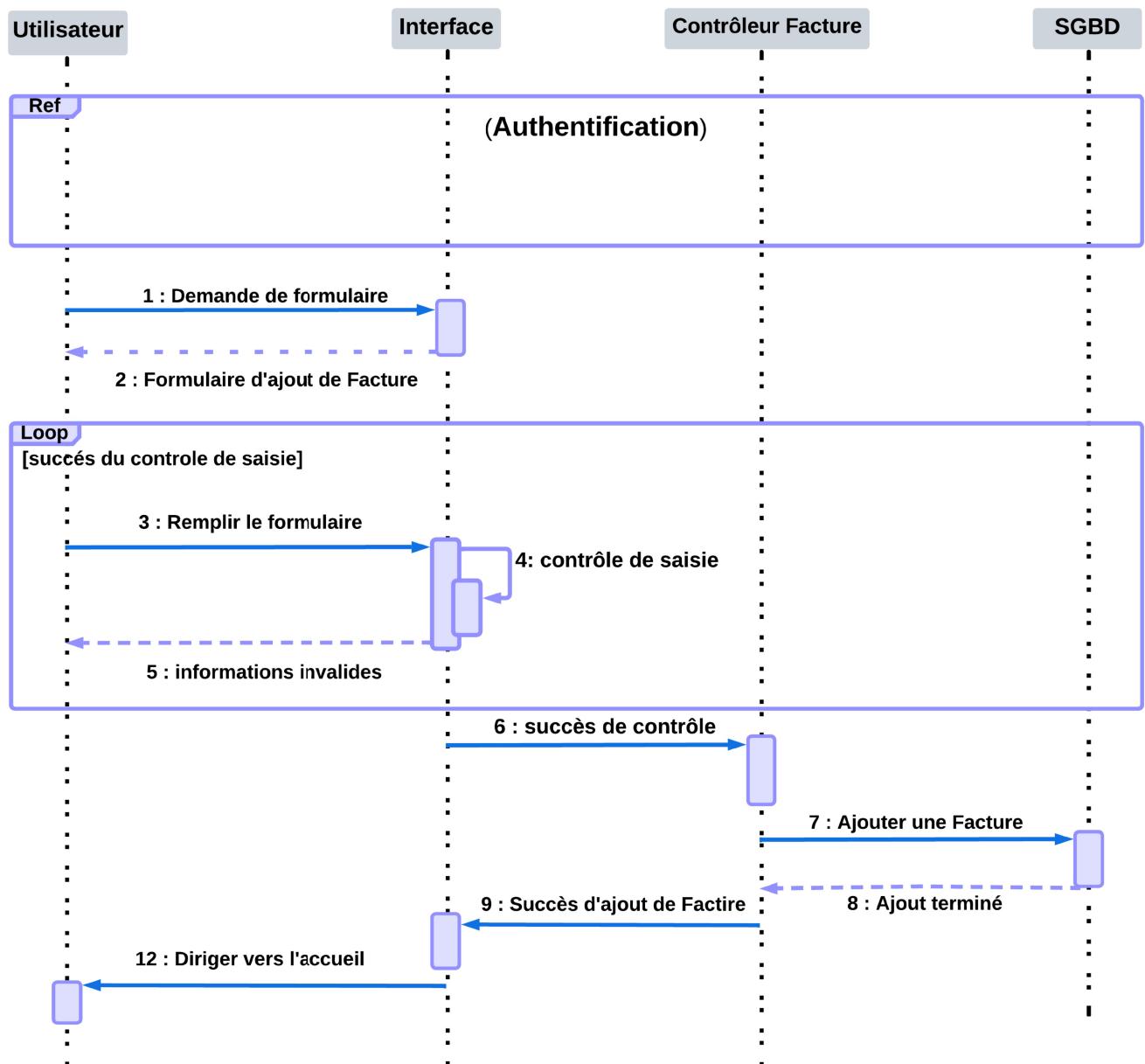


FIGURE 3.3 – Diagramme de séquence du cas d'utilisation « Ajouter une facture »

Pour Ajouter une facture, un administrateur/Employé doit être authentifié pour remplir le formulaire, si les informations saisies sont correctes alors la facture sera ajoutée avec succès . Si les informations sont erronées alors un message d'erreur seront affichées demandant à l'utilisateur de saisir de nouveau les données.

3.3 Vue statique

Le diagramme de cas d'utilisation décrit le système du point de vue acteurs.

Le diagramme de classe décrit la structure interne tout en présentant les différentes classes, leurs attributs, leurs méthodes ainsi que les différentes relations structurelles entre ces classes.

La figure 3.4 représente le diagramme de classes que nous avons utilisé pour développer notre application.

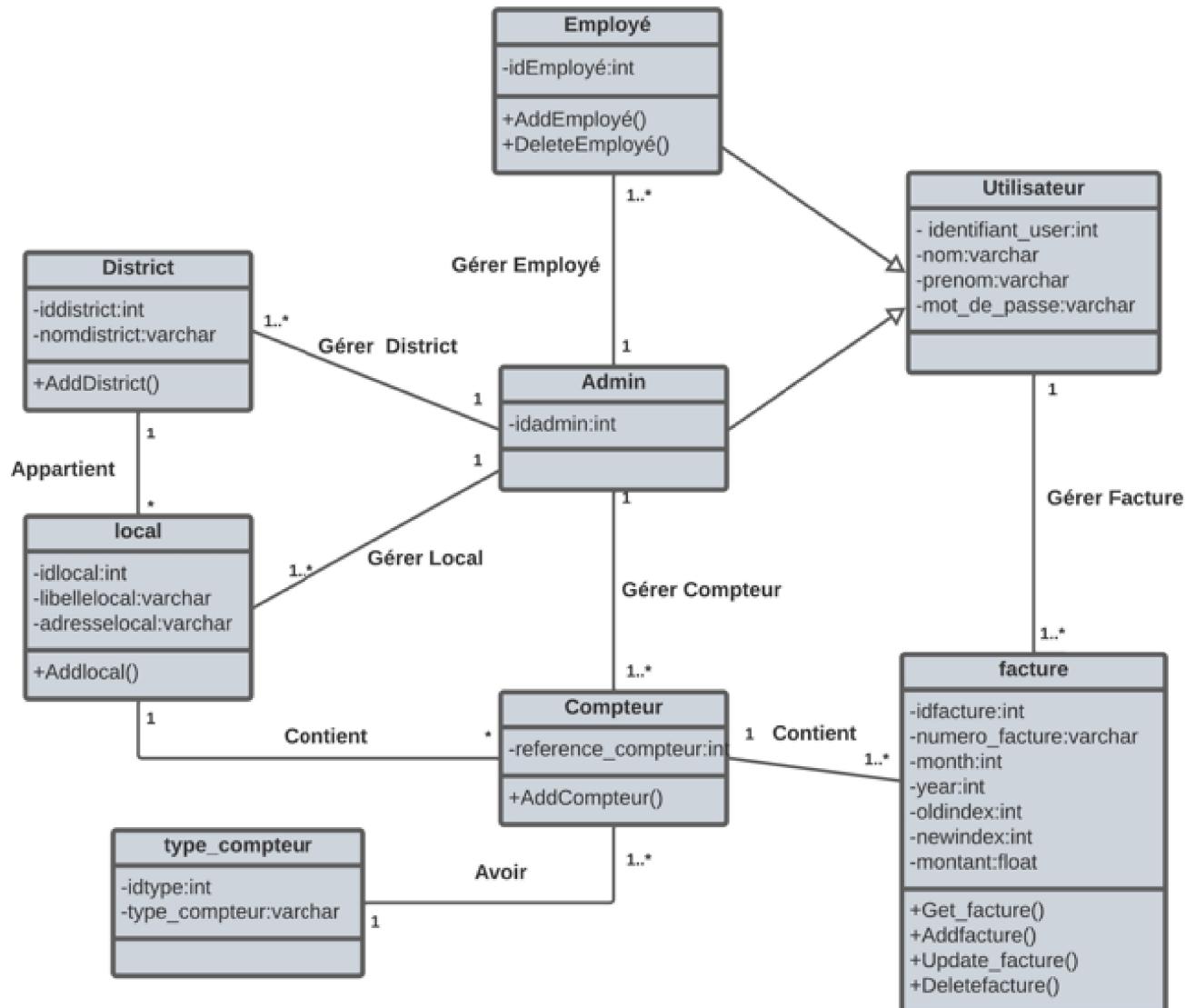


FIGURE 3.4 – Diagramme de Classe

Conclusion

Durant ce chapitre, nous avons décrit l'architecture de notre système et nous avons présenté une vue statique à travers un diagramme de classes décrivant l'organisation du système et une vue dynamique à travers des diagrammes de séquences.

Dans le chapitre suivant, nous allons passer à la réalisation de l'application.

Chapitre 4

Réalisation de L'application

Plan du chapitre

Introduction

1. Environnements du travail
2. Interfaces graphiques de l'application
3. chronogramme de réalisation

Conclusion

Introduction

Après avoir effectué l'étude et la conception de L'application, ce chapitre concerne la phase de réalisation. En d'autres termes, cette dernière section est dédiée aux résultats obtenus par notre application. À cette fin, nous présenterons d'abord l'environnement matériel et logiciel supportant notre Site. Après cela, nous présentons la plateforme de développement et les choix technologiques.

4.0.1 Environnements du travail

Tout au long de la réalisation de notre application, nous avons utilisé des matériels et des logiciels.

4.0.1.1 Environnements de développement matériel

Pour la réalisation du projet, nous avons utilisé un Ordinateur portable Lenovo pour le développement :

Système d'exploitation	Windows 10
disque dur	1 To
Ram	8 Go
Processeur	Intel(R) Core(TM) i3-8145U CPU @ 2.10GHz 2.30 GHz

TABLE 4.1 – caractéristiques de l'ordinateur

4.0.1.2 Environnements de développement logiciel

Dans cette partie,nous nous concentrons aux langages ,aux bibliothèques et aux techniques de programmation utilisées pendant le développement de notre application.

A. IDE de développement :



Visual Studio Code : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Il est multiplateforme, open source et gratuit, supportant une dizaine de langages.

B. SGBD :



MySQL : est un système de gestion de bases de données relationnelles utilisant le langage de programmation SQL. Il propose une version open source qui permet à l'utilisateur d'accéder au code source et de le modifier, et une version entreprise permettant un accès aux dernières fonctionnalités du logiciel et au support fourni par Oracle, propriétaire et développeur actuel de MySQL.

C. Langage de programmation :



Python : est un langage de programmation de haut niveau, interprété et polyvalent, largement utilisé dans le développement logiciel, la science des données, l'automatisation de tâches, la création de sites web et bien d'autres domaines.

D. Bibliothèque : Dans le cadre de ce projet, j'ai choisi de développer l'interface graphique de mon application bureau à l'aide de la bibliothèque Tkinter.

Tkinter : est un module embarqué en Python qui fournit des outils puissants pour créer des interfaces utilisateur conviviales. Tkinter est une bibliothèque bien documentée et largement utilisée pour créer des interfaces graphiques, ce qui la rend idéale pour le développement de mon application. L'un des principaux avantages de Tkinter est sa flexibilité dans la personnalisation de l'interface. J'ai pu créer des fenêtres, des boutons, des zones de texte et d'autres contrôles en quelques lignes de code Python. Cette flexibilité m'a permis de conserver une interface claire et conviviale tout en répondant aux besoins spécifiques de mon application. Tkinter propose également une grande variété d'interactions telles que des boîtes de dialogue, des barres de progression et des événements de clic de souris pour améliorer l'expérience utilisateur.

4.1 Interfaces graphiques de l'application

Les Figuresci-dessous représentent les différentes interfaces de l'application :

4.1.1 Interface d'authentification :

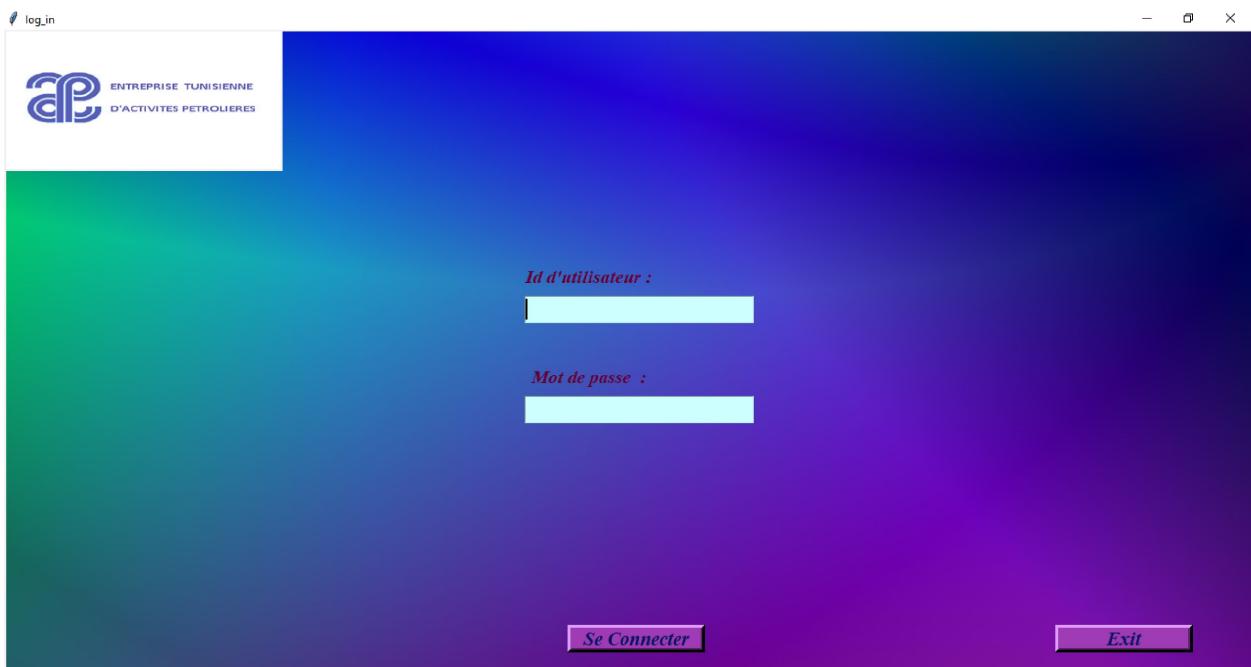


FIGURE 4.1 – Interface : Authentification

4.1.2 L'interface d'accueil

L'employé peut ajouter des factures et il est seulement autorisé à consulter les factures qu'il a ajoutées.



FIGURE 4.2 – Interface : Accueil de l'administrateur

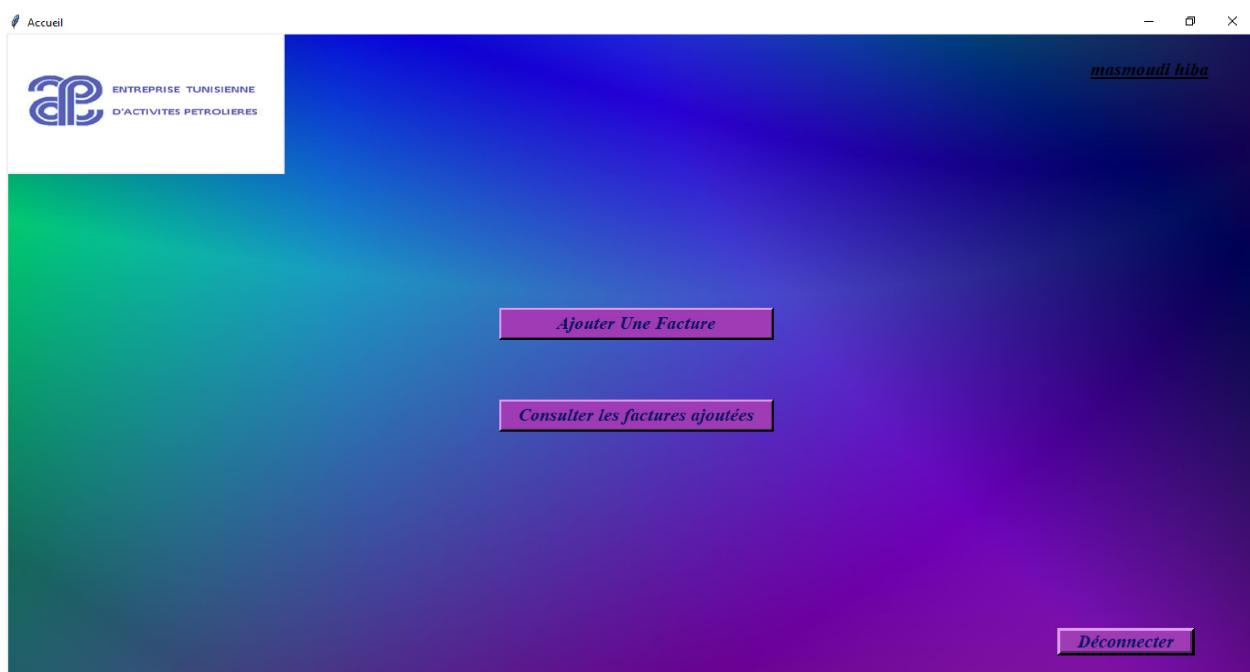


FIGURE 4.3 – Interface : Accueil de l'Employé

4.1.3 Les interfaces d'ajout

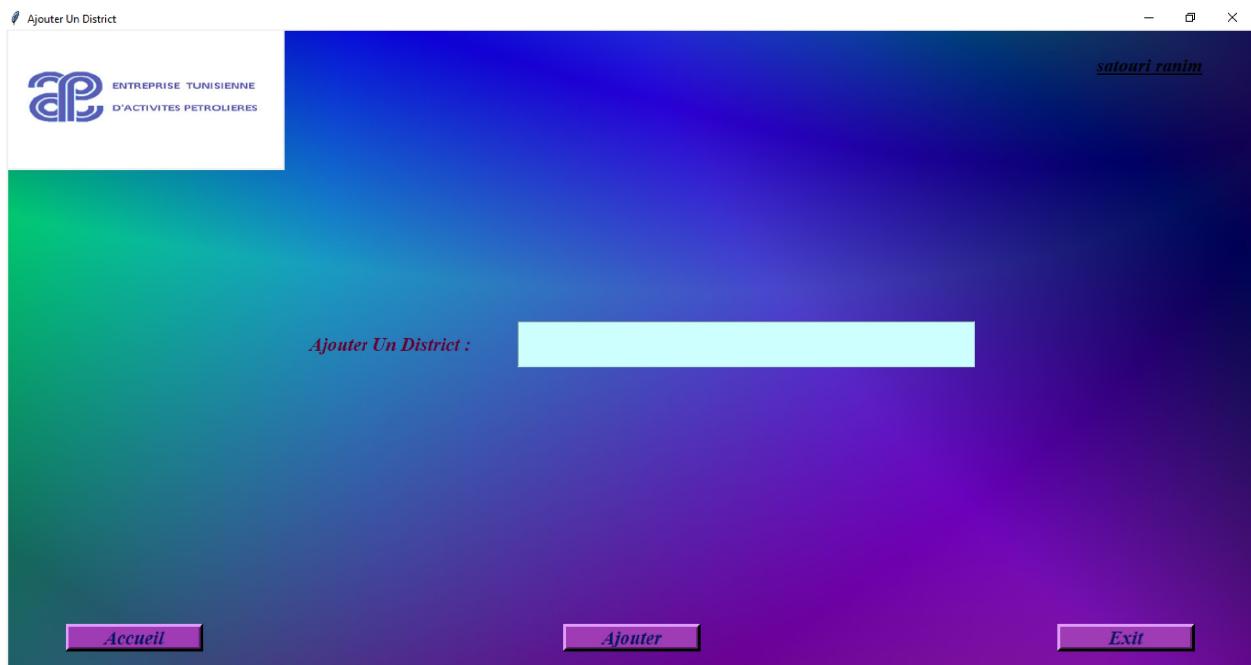


FIGURE 4.4 – Interface : Ajouter un district

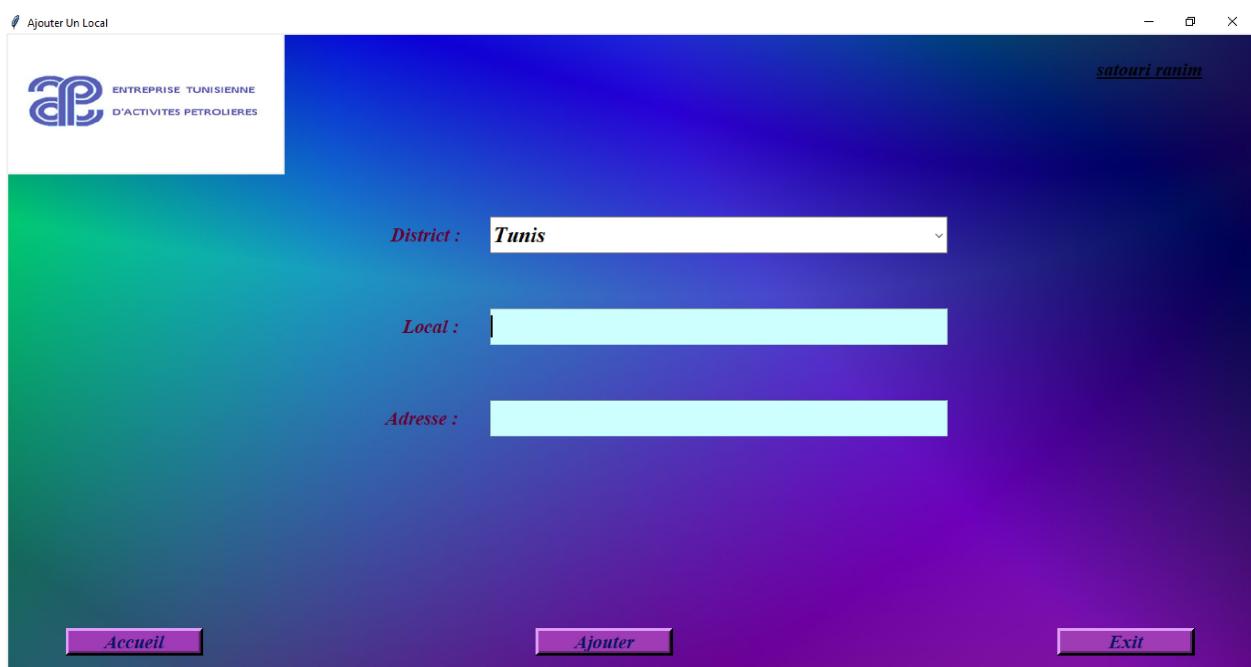


FIGURE 4.5 – Interface : Ajouter un Local

Ajouter Un Compteur

**ENTREPRISE TUNISIENNE
D'ACTIVITES PETROLIERES**

satouri ranim

Local : **le siège (Mohamed V)**

Type Compteur : **Électricité**

Référence :

Accueil **Ajouter** **Exit**

This screenshot shows the 'Ajouter Un Compteur' (Add Meter) interface. At the top left is the logo of 'ENTREPRISE TUNISIENNE D'ACTIVITES PETROLIERES'. On the right is the user name 'satouri ranim'. Below the logo are three dropdown menus: 'Local' set to 'le siège (Mohamed V)', 'Type Compteur' set to 'Électricité', and 'Référence' which is empty. At the bottom are three buttons: 'Accueil', 'Ajouter' (highlighted in blue), and 'Exit'.

FIGURE 4.6 – Interface : Ajouter un compteur

Ajouter une Facture

**ENTREPRISE TUNISIENNE
D'ACTIVITES PETROLIERES**

satouri ranim

District : **benaarous** **OK**

Local : **[empty]** **OK**

compteur : **[empty]**

Numéro De Facture : **[empty]** Montant : **[empty]**

Nouvel Indice : **[empty]**

Mois : **[empty]** Année : **[empty]**

Accueil **Envoyer** **Exit**

This screenshot shows the 'Ajouter une Facture' (Add Bill) interface. It features fields for 'District' (set to 'benaarous'), 'Local' (empty), 'compteur' (empty), 'Numéro De Facture' (empty), 'Montant' (empty), 'Nouvel Indice' (empty), 'Mois' (empty), and 'Année' (empty). At the bottom are three buttons: 'Accueil', 'Envoyer' (highlighted in blue), and 'Exit'.

FIGURE 4.7 – Interface : Ajouter une facture

Ajouter Un Employé

satouri ranim

Nom :

Prénom :

Identifiant :

Mot De Passe :

Accueil **Ajouter** **Exit**

FIGURE 4.8 – Interface : Ajouter un Employé

4.1.4 interface Admin pour gérer les Factures

Consulter Les Factures

satouri ranim

Afficher ALL

District	Libellé Local	Adresse Local	Référence Compteur	Type Compteur	Numéro de Facture	Mois	Année	Ancien Indice	Nouvel Indice	Montant	nom	prenom
Tunis	le siège	Mohamed V	123456	electricité	654321	2	2023	0	574123	7458.0	satouri	ranim
Tunis	le siège	Mohamed V	968574	eau	253614	2	2023	0	452136	5824.0	satouri	ranim
Tunis	le siège	Mohamed V	123456	electricité	541287	5	2023	0	547821	2354.54	masmoudi	hiba
Tunis	le siège	Mohamed V	365214	gaz	254178	7	2023	547123	580001	2354.0	masmoudi	hiba
Tunis	Immeuble Zouila Montplaisir		524168	eau	25874	8	2023	0	256987	452.0	ben salah	nour
Tunis	Centre de reche	charguia	574821	eau	25478	5	2023	0	52147	2305.0	satouri	ranim
Tunis	le siège	Mohamed V	968574	eau	74521	6	2023	0	63254	7125.0	satouri	ranim
Tunis	Centre de reche	charguia	574821	eau	25478	7	2023	52147	65213	2541.0	satouri	ranim
Tunis	Centre de reche	charguia	652874	gaz	25417	2	2023	0	25417	2354.0	masmoudi	hiba
Tunis	Centre de reche	charguia	541236	electricité	12547	6	2023	0	32564	1251.0	ayari	anwer
Tunis	Centre de reche	charguia	541236	electricité	54782	7	2023	0	36524	2547.0	satouri	ranim

Accueil **Modifier** **supprimer** **Exit**

FIGURE 4.9 – Interface : Consulter les Factures

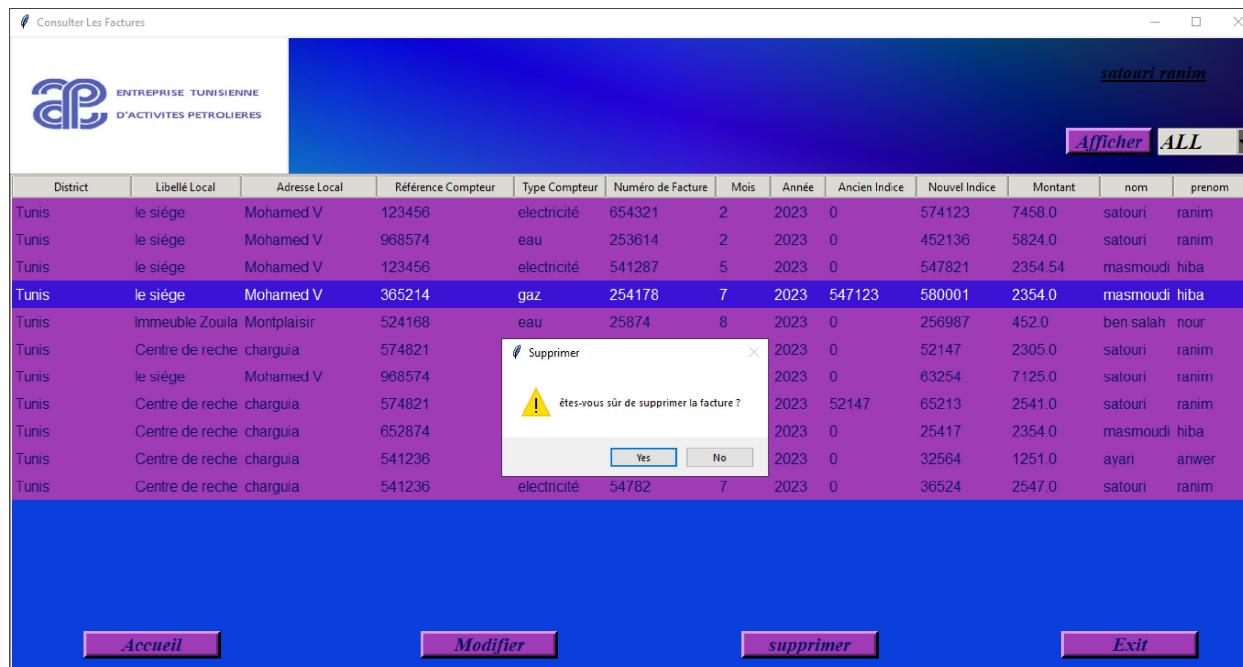


FIGURE 4.10 – Interface : Confirmer la suppression d'une facture

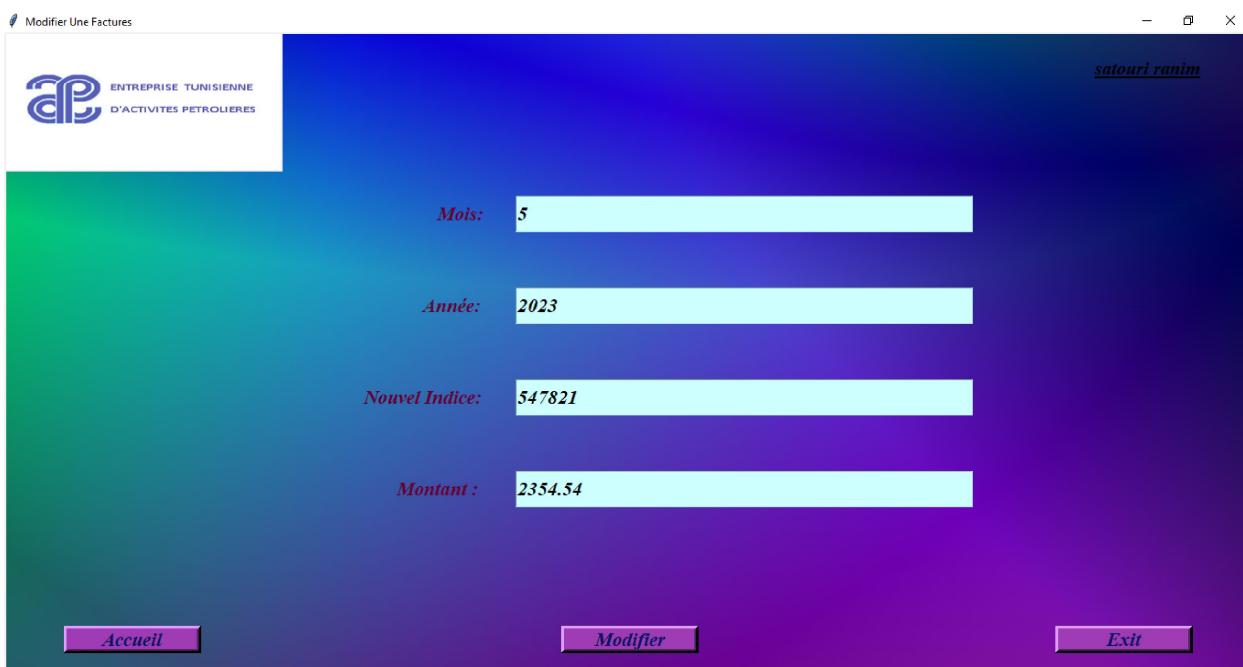


FIGURE 4.11 – Interface : Modifier une facture

4.1.5 Interface admin pour gérer les Employés

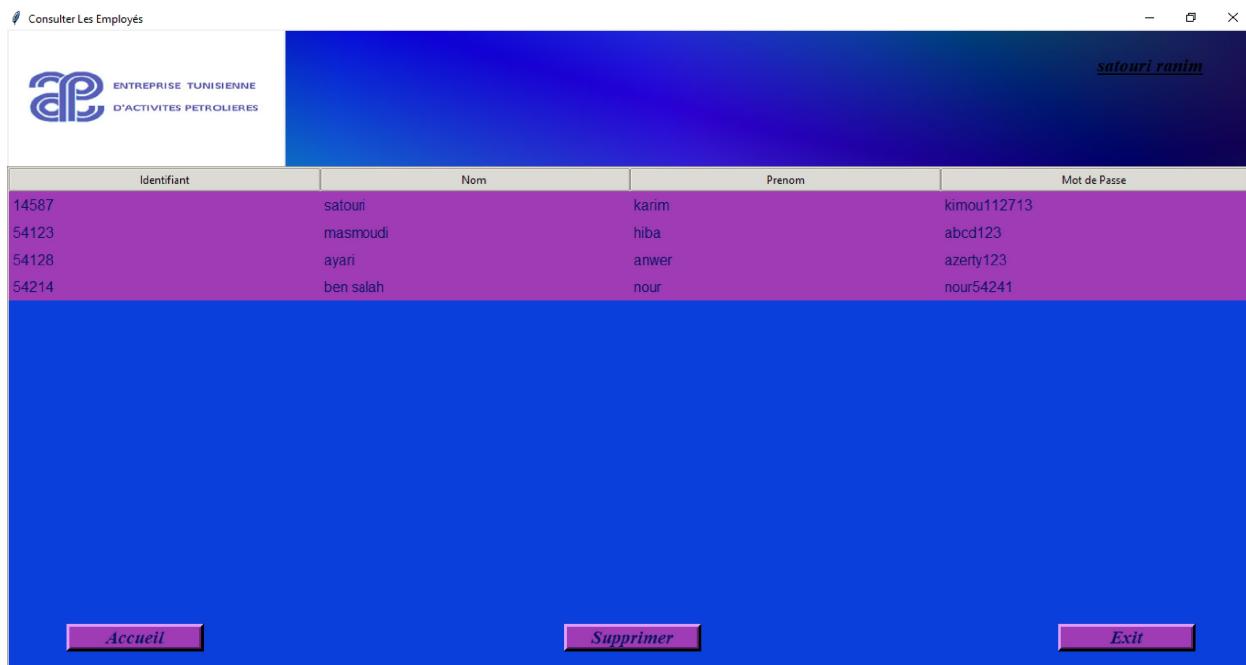


FIGURE 4.12 – Interface : consulter les Employés

4.1.6 Interface d'états statistiques

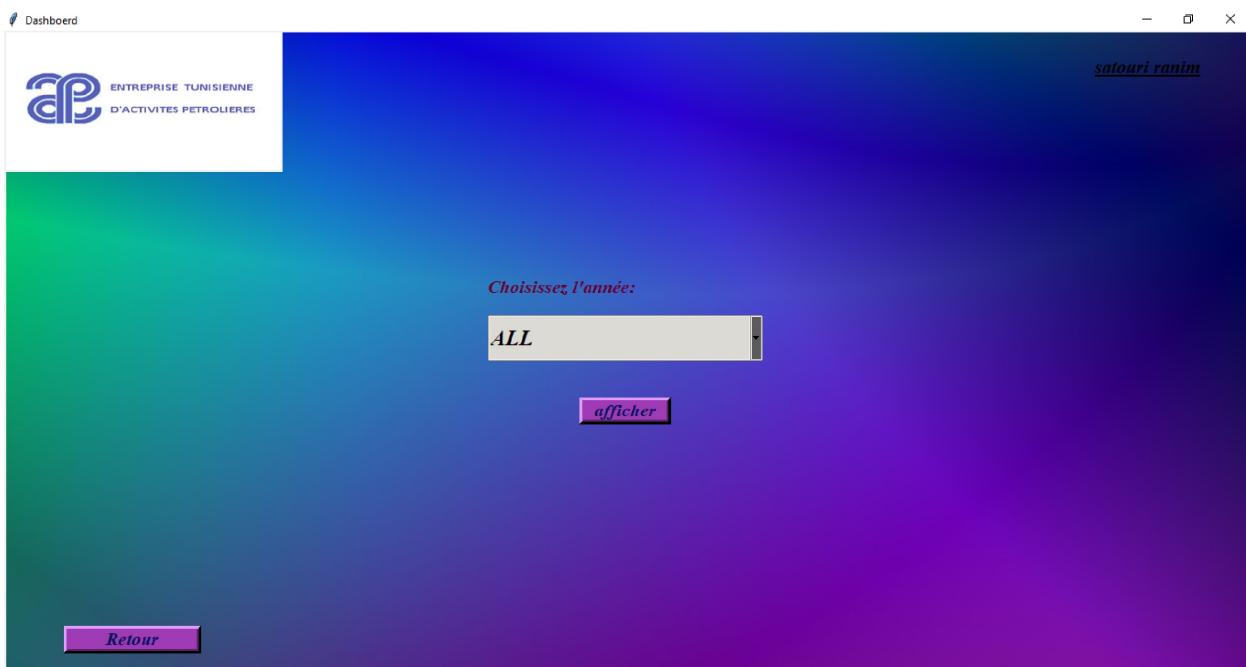


FIGURE 4.13 – Interface : selectionner l'année

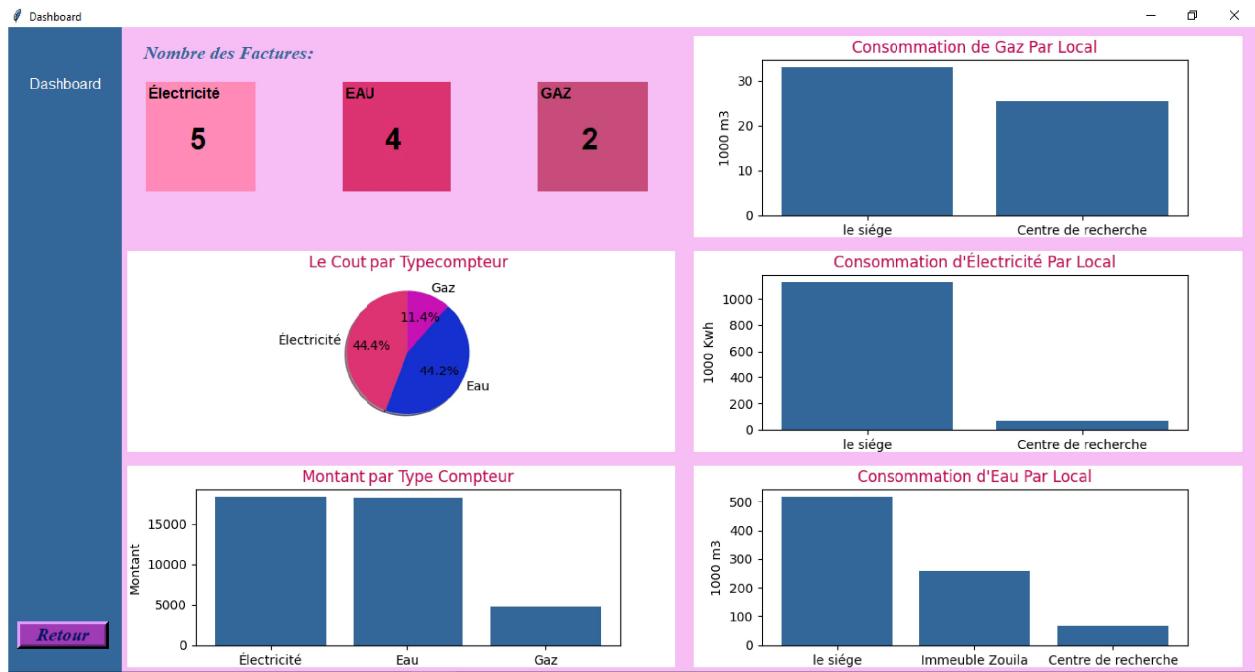


FIGURE 4.14 – Interface : dashboard

4.2 Chronogramme de réalisation du projet

Le diagramme de Gantt est un outil utilisé en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet. Il permet de représenter graphiquement l'avancement de ce dernier. Pour le bon déroulement de notre projet, la figure présente notre diagramme de Gantt, schématisé en respectant le planning de réalisation de nos tâches tout au long de la période de notre stage, présenté dans le tableau 4.2.

Nom de tache	Durée	Début	Fin
Spécification des besoins	3 jours	01/08/2023	03/08/2023
Conception	5 jours	04/08/2023	08/08/2023
Développement	23 jours	09/08/2023	31/08/2023
Rapport	10 jours	22/08/2023	31/08/2023

TABLE 4.2 – Planning de réalisation des tâches tout au long de la période de stage

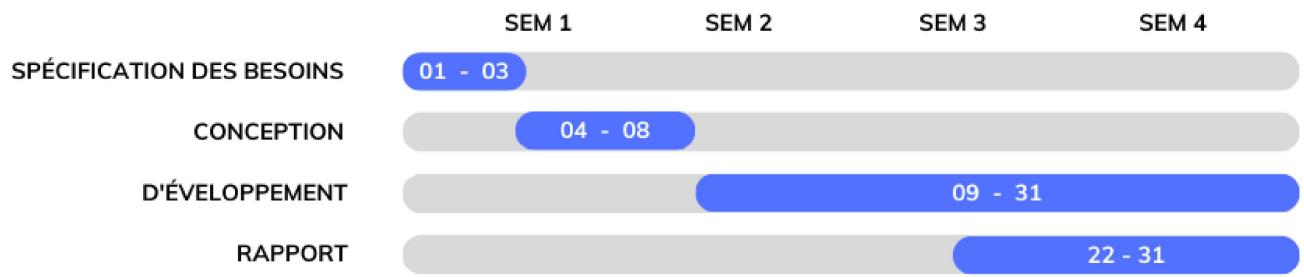


FIGURE 4.15 – Diagramme de Gantt

Conclusion

A travers ce chapitre, nous avons clôturé notre projet en présentant des choix techniques. Nous avons exposé le travail effectué à travers les différentes interfaces apparues lors de l'exécution du projet.

Conclusion générale et perspectives

Ce projet de gestion des facturation développé au sein de l’Entreprise Tunisienne de l’Activité Pétrolière (ETAP) a été une opportunité exceptionnelle à plusieurs égards. Tout d’abord, il a marqué le début de mon parcours en tant qu’étudiant en génie logiciel à la Faculté des Sciences, en me permettant de mettre en pratique les connaissances et compétences acquises au cours de ma première année. En tant que projet d’initiation , il m’a ouvert les portes du monde professionnel de l’informatique, même si ce n’était que pour une courte période d’un mois.

En tant qu’étudiant en informatique, ce stage d’été a également été l’occasion d’explorer de nouvelles technologies et de renforcer mes compétences. J’ai eu la chance d’apprendre le langage SQL et de travailler avec MySQL pour gérer la base de données du projet. J’ai également plongé dans le monde de la conception logicielle en utilisant UML (Unified Modeling Language) pour modéliser l’architecture de l’application. De plus, j’ai découvert la bibliothèque Tkinter, une toute nouvelle expérience pour moi, en créant une interface utilisateur conviviale pour l’application. Mon expérience m’a également permis d’améliorer ma programmation en Python et d’apprendre à gérer un projet de développement de logiciel de A à Z.

En rétrospective, je considère que ce projet a été une étape importante dans mon parcours académique et professionnel. Bien que ma première année d’études ne m’ait pas encore exposé à de nombreux concepts avancés, j’ai abordé ce projet avec détermination et la volonté d’apprendre par moi-même. J’ai découvert la satisfaction de relever des défis techniques et de les surmonter grâce à l’auto-apprentissage.

En envisageant l’avenir de ce projet, je crois fermement qu’il existe des opportunités d’amélioration significatives. Une telle amélioration pourrait consister à intégrer la technologie OCR (Optical Character Recognition) afin de scanner automatiquement les factures, éliminant ainsi la saisie manuelle des données. Cette avancée serait une étape majeure vers l’automatisation complète du processus de gestion des factures, permettant ainsi de gagner du temps et d’accroître l’efficacité.