**Dédicaces**

Nous dédions ce modeste travail, comme preuve de respect, de  
gratitude, et de reconnaissance à :

Nos chers parents : source de tendresse de noblesse et d’affectation qui n’ont jamais cessé de nous encourager et  
nous conseiller.

Nos chers frères et sœurs : Pour leurs encouragements permanents et leur soutien moral.   
Nos amis et collègues : Pour leur compagnie et bons moments passés ensemble.

Tous les professeurs qui nous donnent nos premières  
connaissances.  
  
Merci d’être toujours là pour nous

Amine et Donia

**Remerciements**

Avant de commencer le développement de cette expérience professionnelle, il nous parait tout naturel de commencer par remercier les personnes qui nous ont permis d’effectuer ce travail ainsi que ceux qui nous ont permis d’en faire un moment agréable et profitable.

Nous remercions donc Mr Imed Zaier, de nous avoir fait confiance et ainsi de nous offrir la possibilité de vivre cette expérience professionnelle, à Mme Aicha Ben Salem de nous avoir formé avec rigueur et patience .Et, pour terminer, nous voulons remercier l’ensemble des employés sur toute aide qu’ils ont pu nous prodiguer au cours de cette expérience.

**Table des matières**

Dédicace

Remerciement

Introduction générale

**Chapitre1 : Cadre général du projet**

Introduction

1. Présentation de l’organisme d’accueil

1.1 Centre National Informatique CNI

1.2 Description du CNI

1.3. Problématique

1.4. La solution proposée

Conclusion

**Chapitre2 : Analyse et spécification des besoins**

Introduction

1. Description des acteurs

2. Spécifications des besoins

2.1 Les besoins fonctionnels

2.2 Les besoins non fonctionnels

3. Méthodologies du travail

4. Diagrammes des cas d’utilisations

4.1. Diagrammes des cas d’utilisation global

4.2. Diagrammes du cas d’utilisation « Gérer les comptes des utilisateurs »

4.3. Diagrammes du cas d’utilisation « Gérer les contrats »

Conclusion

**Chapitre3 : Conception**

Introduction

1. Architecture de l’application
   1. Architecture MVC
   2. Avantages de l’architecture
2. Les vues de la conception
   1. Vue statique : Le diagramme de classe
   2. Vue dynamique : Les diagrammes de séquences
   3. Diagramme de séquence « S’authentifier»

2.4. Diagramme de séquence « Modifier un utilisateur »

2.5. Diagramme de séquence « Ajouter un contrat »

2.6. Diagramme de séquence « Supprimer un contrat

2.7. Diagramme de séquence « Consulter un contrat »

Conclusion

**Chapitre4 : Développement et réalisation**

Introduction

1. Environnement du travail

1.1 Environnement du travail matériel

1.2 Environnement du travail logiciel

1.2.1. Les langages de programmation

1.2.2. Les Framework

1. Interface graphique

2.1. Interface « Authentification »

2.2. Interface « «administrateur »

2.2.1. Ajouter un compte utilisateur

2.2.2. Modifier un compte utilisateur

2.2.3. Supprimer un compte utilisateur

2.3. Interface « Responsable des contrats »

2.3.1. Ajouter un contrat

2.3.2. Modifier un contrat

2.3.3. Supprimer un contrat

2.3.4. Renouveler un contrat

2.3.5. Consulter un contrat

2.3.6. Imprimer un contrat

2.3.7. Rechercher un contrat

Conclusion

Conclusion Générale

Références

**Liste des figures**

**Figure 1: logo du CNI**

**Figure 2: Méthodologie en cascade**

**Figure 3: diagramme des cas d'utilisations général**

**Figure 4: diagramme du cas d'utilisation «Gérer les comptes utilisateurs »**

**Figure 5: diagramme du cas d'utilisation « Gérer les contrats»**

**Figure 6: Architecture en MVC**

**Figure 7: Le diagramme de classe**

**Figure 8: Diagramme de séquence « S’authentifier »**

**Figure 9: Diagramme de séquence « Modifier un utilisateur »**

**Figure 10: Diagramme de séquence « Ajouter un contrat »**

**Figure 11: Diagramme de séquence « Supprimer un contrat »**

**Figure 12: Diagramme de séquence «  Consulter un contrat »**

**Figure 13: Logo de Visual Studio Code**

**Figure 14: Logo de XAMPP**

**Figure 15: Logo de Notepad++**

**Figure 16: Logo d’UML**

**Figure 17: Logo de HTML**

**Figure 18: Logo de CSS**

**Figure 19: Logo de Node.js**

**Figure 20: Logo d’Angular**

**Figure 21: Logo de Bootstrap**

**Liste des tableaux**

**Tableau 1 : Description textuelle du diagramme des cas d’utilisations global**

**Tableau 2 : Description textuelle du cas d’utilisation «Gérer  les comptes des utilisateurs »**

**Tableau 3 : Description textuelle du cas d’utilisation « Gérer  les contrats»**

**Tableau 4 : Description de la table « Utilisateur »**

**Tableau 5 : Description de la table « Client »**

**Tableau 6 : Description de la table « Fournisseur »**

**Tableau 7 : Description de la table « Administrateur »**

**Tableau 8 : Description de la table « Responsable des contrats client »**

**Tableau 9 : Description de la table « Responsable des contrats fournisseurs »**

**Tableau 10 : Description de la table « Contrat client »**

**Tableau 11 : Description de la table « Contrat fournisseur »**

**Tableau 12 : Description textuelle du diagramme de séquence « S’authentifier »**

**Tableau 13 : Description textuelle du diagramme de séquence « Modifier un utilisateur »**

**Tableau 14 : Description textuelle du diagramme de séquence « Ajouter un contrat »**

**Tableau 15 : Description textuelle du diagramme de séquence « Supprimer un contrat»**

**Tableau 16 : Description textuelle du diagramme de séquence « Consulter un contrat »**

**Introduction générale**

Dans les années 90 à 2000, les entreprises ont fait d’énormes investissements informatiques pour leurs systèmes transactionnels et les progiciels de gestion intégrés. Le constat d’aujourd’hui est que pour ce qui est de la planification et de la gestion, Excel est, depuis bon nombre d’années, omniprésent. **Plus de 80% des utilisateurs utilisent Excel.** Mais si cet outil est très flexible et permet à chaque personne au sein de l’entreprise de manipuler ces données et d’essayer de faire des prévisions, sa réalité au quotidien est qu’il ne permet pas du tout de gérer de gros volumes de données, et surtout qu’**il n’est absolument pas collaboratif**. Avec Excel, chacun possède son propre fichier et les entreprises passent énormément de temps à réconcilier ces informations et ces fichiers entre eux pour en faire un master Excel qui au final ne fournit pas d’informations correctes ni fiables.

C’est un outil qui reste complexe à manipuler et enfin, la plupart du temps, ces tableurs comportent des erreurs de calcul que personne ne vient contrôler : les vérifier provoquerait trio de débats houleux et prendrait aussi trop de temps. En fin de compte, on abandonne la partie et les gens « font confiance ».

Dans un monde de plus en plus imprévisible et qui impose aux entreprises qui veulent rester compétitives, de pouvoir prendre des décisions au plus vite**, l’usage d’Excel est devenu un fléau contemporain.**

L’informatique de gestion correspond généralement à de nombreuses tâches effectuées dans le cadre d’une entreprise. Le but est en effet d’organiser des informations, de les vérifier, de les collecter, de manière à être plus productif. L’informatique a beaucoup apporté à la gestion des données au sens large, en faisant gagner un temps précieux, notamment aux professionnels qui aujourd’hui ne pourraient tout simplement plus se passer de ce type de dispositif.

De nos jours, rares sont les entreprises qui n’ont pas de stratégie digitale, pour faire la différence, il faut non seulement avoir une application web, mais aussi que cette dernière soit bien placée dans les moteurs de recherche. Les habitudes du public ont changé, peu importe le bien ou le service demandé, c’est sur le Web que se fait la recherche. Pour ne pas laisser la chance filer, l’entreprise doit avoir une présence digitale. En d’autres termes, il faut être bien présent et visible sur Internet et cela commence par la création d’une application Web. Il ne s’agit plus seulement d’être une entreprise physique, il faut aussi être une entreprise digitale, une entreprise connectée, une entreprise visible et médiatisée.

**Chapitre 1 :**

**Cadre général du projet**

**Introduction**

Ce chapitre est consacré à l'organisme d'accueil, à ses différents services et aux technologies employées au sein de l'entreprise. Il décrit également l'objectif du stage.

**1. Présentation de l’organisme d’accueil**

Dans ce chapitre, il s'agit de représenter les établissements d'accueil et leurs activités, d'examiner les modes actuels de gestion et suivi des contrats et des conventions et de préciser l'importance du développement web dans la vie professionnelle.

**1.1 Centre Nationale de l’Informatique CNI**

Le Centre National de l’Informatique est un établissement public à caractère non administratif doté, de la personnalité civile et de l’autonomie financière, institué le 30 Décembre 1975. Le CNI est un organisme placé sous la tutelle du Ministre des Technologies de la Communication et de l’Economie Numérique, et opérant dans les domaines du secteur de l’informatique et des technologies de la communication et certifié ISO 9001 versions 2015. C’est un principal appui aux structures publiques de l’administration dans la réalisation, le déploiement et l’exploitation des systèmes d’information.



Figure 1:Logo du CNI

**1.2. Description du CNI**

Le CNI assure les activités suivantes :

-Le CNI est l’opérateur national qui assure l’interopérabilité des SI de l’administration tunisienne.

-Le CNI est le Cloud national.

-Maitrise d’ouvrage déléguée et pilotage de projet

-Etudes et Conseil

-Développement

-Hébergement du système

-Backup et continuité d’activité

-Formation et certification

-Déploiement des applications et assistance des utilisateurs

**1.3. Problématique :**

La gestion et le suivi des contrats au sein du CNI se fait par un fichier Excel.

La gestion des contrats au sein CNI face plusieurs difficultés qui sont les suivants :

* une absence de vision globale sur les données.
* Excel n’est pas un outil dynamique : Les données ne s’actualisent pas automatiquement.
* Excel n’est pas un outil interactif :

Sur Excel, pas de notifications, ni d’alertes pour vous informer des dernières modifications sur un document

### Excel n’est pas un outil sécurisé.

* Excel n’est pas un outil collaboratif :

Si vous travaillez sur un fichier Excel et qu’un autre membre de l’équipe souhaites-y apporter des modifications, cela va générer un nouveau fichier car il est impossible de travailler simultanément sur un même document Excel.

* Excel n’est pas un outil de suivi :

Le fait qu’Excel ne soit pas un outil dynamique pose également un problème au niveau du suivi du projet, si le fichier Excel n’est pas mis à jour quotidiennement, difficile pour le chef de projet et son équipe de savoir exactement quel est l’avancement du projet.

**1.4. La solution proposée :**

La gestion des contrats au sein du CNI est constamment confrontée à de nombreuses difficultés. Pour faire face à cette situation, on propose une solution applicative qui permettrait aux utilisateurs d'être informés des informations nécessaires sur les contrats (état du contrat…).

Le projet consiste à concevoir, faire et développer une application qui permet de gérer les contrats en ligne avec un tableau de bord.

Il s’agit de définir les responsabilités de la gestion, mettre à jour les données, organiser des données collectées, renforcer le contrôle et la confrontation, garantir une meilleure gestion des contrats, assurer une cohérence d’information et faciliter et simplifier le travail des responsables.

Cette solution apporte les avantages suivants :

• Automatisant et rationalisant les processus d’examen et d’approbation des contrats.

• Immatérialisions des contrats

• Disponibilité des contrats

• Centralisation de l’information

• Économie de l’argent

• Gain du temps

• Donnant aux personnes autorisées la possibilité de trouver, d’accéder et d’éditer les contrats plus facilement et plus rapidement.

• Contrôlant la réalisation de toutes les étapes de la contractualisation.

Cette application vise à:

* Présenter les informations relatives à chaque contrat d'achat, maintenance (fournisseurs) et vente (clients).
* Informer sur le cycle de vie des contrats à travers une indication de couleurs (chaque couleur représente l'état de contrat).

**Conclusion :**

Tout au long de ce chapitre nous avons présenté l'organisme d'accueil et les besoins nécessaires à la réalisation de notre projet en passant par plusieurs étapes commençons par l'étude de l'existant tout en comprenant les besoins de l’entreprise et déterminant les problématiques actuelles afin de proposer les solutions adéquates pour la réalisation de nos objectifs.

**Chapitre2 :**

**Analyse et spécification des besoins**

**Introduction :**

L'analyse et la spécification des besoins représentent la première phase du cycle de développement d'un logiciel. Elle sert à identifier les acteurs réactifs du système et leur associer chacun l'ensemble d'actions avec lesquelles il intervient dans l'objectif de donner un résultat optimal et satisfaisant au client.

La spécification des besoins représente l'ensemble des services que doit fournir le logiciel à son utilisateur. Selon le processus unifié chaque service est modélisé sous un cas d'utilisation, pour élaborer à la fin un diagramme UML de cas d'utilisation.

Chaque cas d'utilisation est décrit sous une forme textuelle représentant un scenario nominal (ensemble des actions à réaliser pour atteindre l'objectif).

**1. Identifications des acteurs**

Nous allons maintenant énumérer les acteurs susceptibles d'interagir avec le système. Tout d'abord, nous commençons par définir ce qui est un acteur.

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Les acteurs de notre application sont :

L'administrateur :

Le responsable des contrats client :

Le responsable des contrats fournisseurs

**2. Spécifications des besoins:**

La spécification des besoins consiste à dégager tous les acteurs en les décrivant et d’exprimer dans les phrases les principales actions du système et des acteurs, elle permet aussi de représenter et décrire le diagramme de cas d'utilisation pour donner une vue externe sur notre système.

### 2.1. Besoins Fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins qui spécifient un comportement d'entrée / sortie du système.

En effet, ces besoins répondent aux points précis du cahier de charge et prennent en considération surtout les besoins de l'utilisateur.

Les fonctionnalités offertes par notre application sont:

* La gestion des contrats clients (vente) et les contrats fournisseurs(Achat).
* Le suivi des utilisateurs : L’ajout, la modification et la suppression des utilisateurs du système par l’administrateur.
* Le stockage et l’organisation des données.
* L’accès en temps réel aux données.
* L’authentification.

### 2.2. Besoins Non Fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception.

Le système doit vérifier les contraintes liés à l'implémentation (langage de programmation, de système d'Exploitation...) ou à l'interopérabilité générale (ne pas bouffer toutes les ressources de la machine) afin de satisfaire les besoins de l’utilisateur.

Parmi ces besoins :

* **La fiabilité :**

Le délai de réponse en temps réel de toutes les fonctionnalités attendues doit être rapide, en effet l’utilisateur ne doit pas remarquer une lenteur pendant l’accès à l’information ou lors de l’utilisation de l’application.

* **La sécurité d’accès :**

La phase d’authentification permet à chaque utilisateur d’accéder seulement aux fonctionnalités attribuées à son domaine.

* **L’extensibilité :**

L’application doit être évolutive permettant de faire des modifications à tout moment.

Ceci va permettre d’adapter le système aux futurs besoins des utilisateurs.

* **L’ergonomie des interfaces :**

L’utilisateur aura le droit d’avoir une interface simple et pratique pour qu’il puisse utiliser l’application sans aucune lacune, en effet l’application doit être facile à manipuler et à comprendre.

* **La performance :**

Le système doit être capable à répondre dans des délais réduits et gérer les interruptions causées par les erreurs de manipulation.

* **La portabilité:**

C’est la capacité de l’application à pouvoir d’être adaptée facilement en vue de fonctionner dans différents environnements d’exécution.

**3. Méthodologie du travail :**

Vu que les besoins auxquels doit répondre notre application sont stables et clairement identifiés dès le début, nous avons choisi comme modèle de cycle de vie de l’application, le modèle en cascade, jugé le plus adapté pour la conception de note projet.

Le modèle en cascade offre une **structure hiérarchique claire** pour les projets de développement dans laquelle les différentes phases du projet sont clairement délimitées. Étant donné que chaque phase se termine par une étape, le processus de développement peut être suivi facilement.

La méthode en cascade se compose de **cinq étapes :**

* **Analyse :** planification, analyse et spécification des besoins
* **Conception :** conception et spécification du système
* **Implémentation :** programmation et tests des modules
* **Test :** intégration du système, tests du système et de l’intégration
* **Exploitation :** livraison, maintenance, amélioration

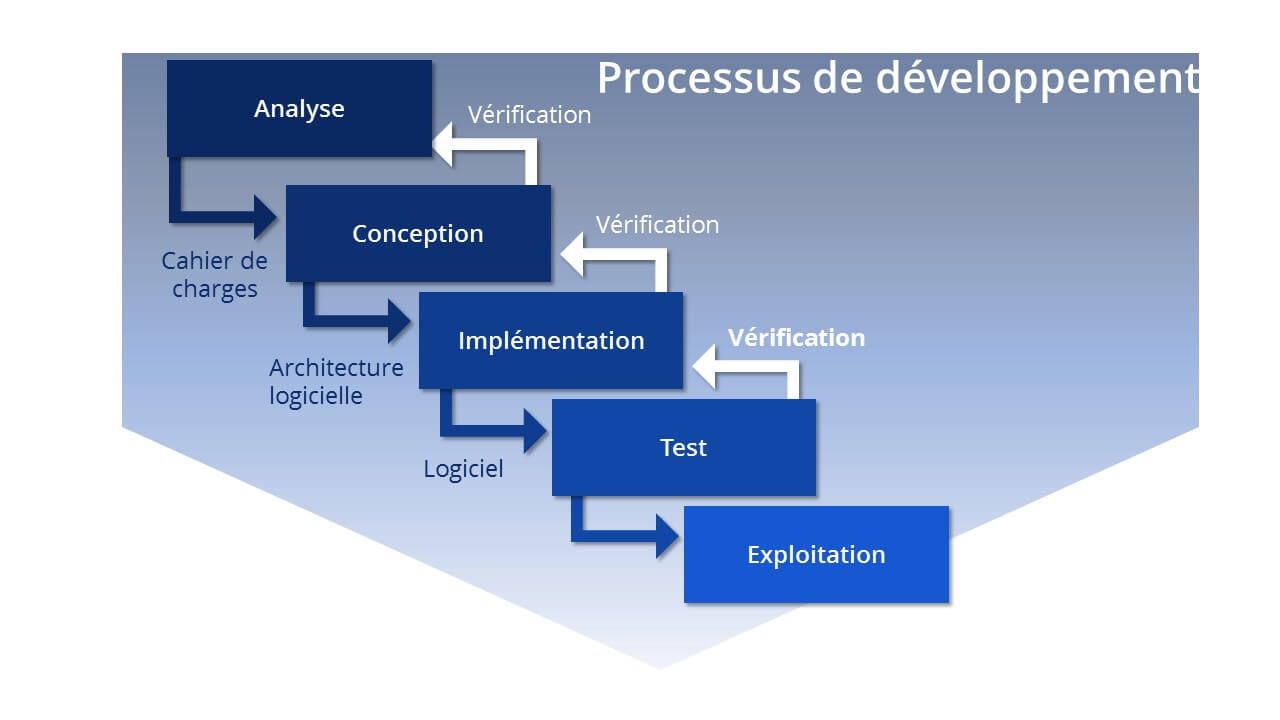
****

Figure 2: Méthodologie en cascade

### 4. Les diagrammes de cas d’utilisations :

Les diagrammes de cas d'utilisations décrivent sous la forme d'actions et de réactions le comportement du système du point de vue utilisateurs .ils permet de définir les limites et les relations entre le système et son environnement et ils recouvrent toutes les phases de développement depuis le cahier des charges jusqu'à l’implémentation.

Un cas d'utilisation représente l'utilisation et le fonctionnement du système.

**4.1. Diagramme de cas d'utilisation général :**

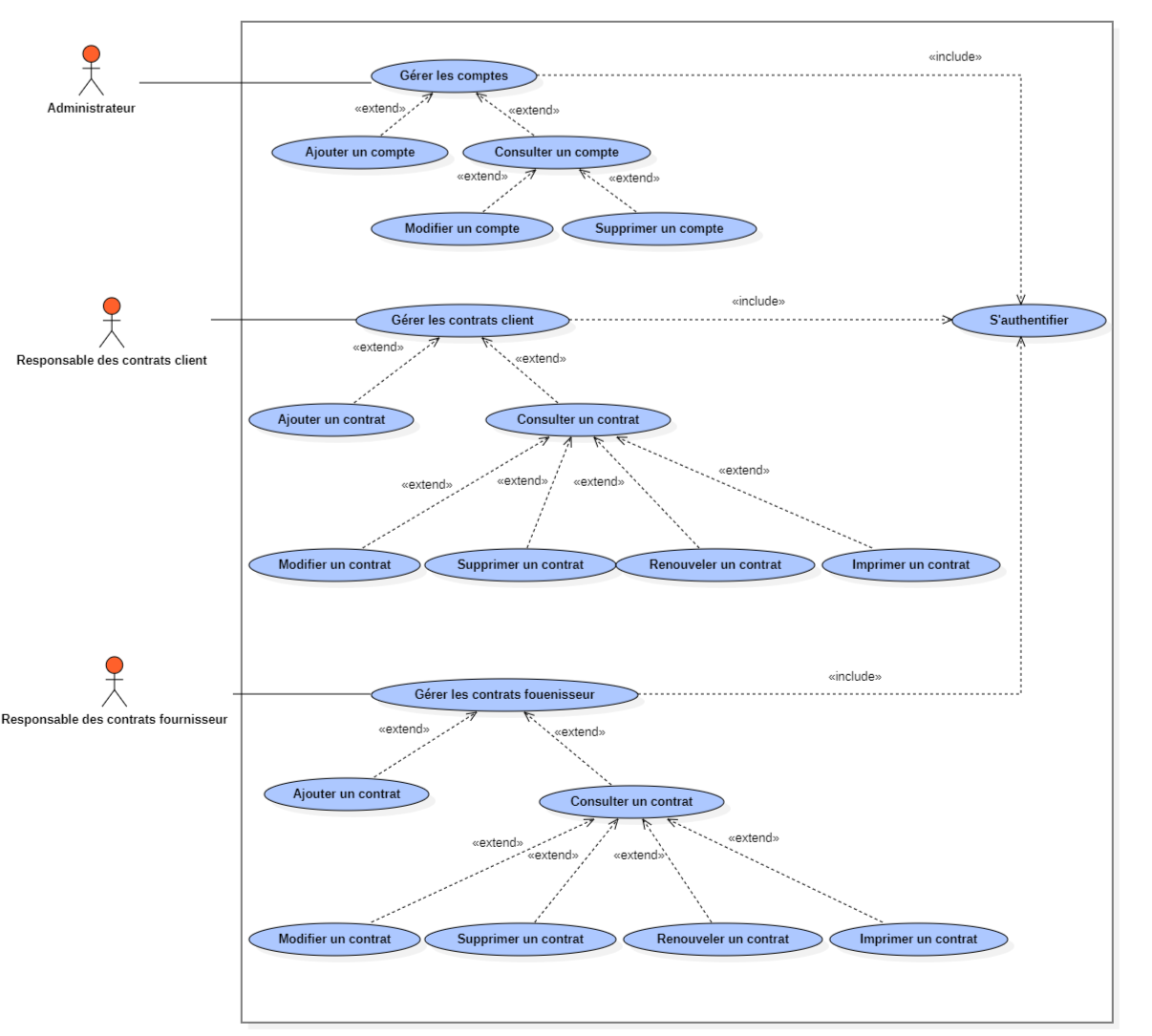


Figure 3 : Diagramme de cas d'utilisation global

Dans notre projet le diagramme ci -dessus représente les différents cas d’utilisations offertes à l'utilisateur qui sont les suivants :

|  |  |
| --- | --- |
| Pour l'administrateur : | Pour le responsable des contrats clients/fournisseurs : |
| \*Ajouter un compte.  \*Modifier un compte.  \*Supprimer un compte | \*Ajouter des contrats  \*Modifier des contrats  \*Supprimer des contrats  \*Consulter des contrats  \*Renouveler des contrats  \*Imprimer les contrats |

**Tableau 1 : Description textuelle du cas d’utilisation global**

**4.2. Diagramme du cas d’utilisation « Gérer les comptes des utilisateurs »**

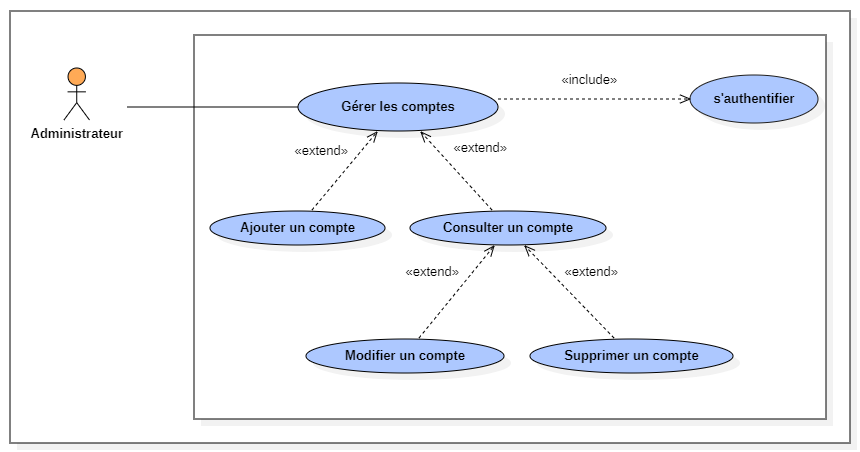


Figure 4 : Gérer les comptes des utilisateurs

**Description textuelle du cas d’utilisation « Gérer les comptes des utilisateurs »**

|  |
| --- |
| Titre : Gérer les comptes des utilisateurs |
| Acteur : Administrateur |
| Pré condition : S’authentifier |
| Post condition : Mise à jour des comptes |
| Les cas possibles :  \*Ajouter un compte  \* Modifier un compte  \* Supprimer un compte |

**Tableau 2 : Description textuelle du cas d’utilisation « Gérer les comptes des utilisateurs »**

**4.3. Diagramme du cas d’utilisation « Gérer les contrats»**

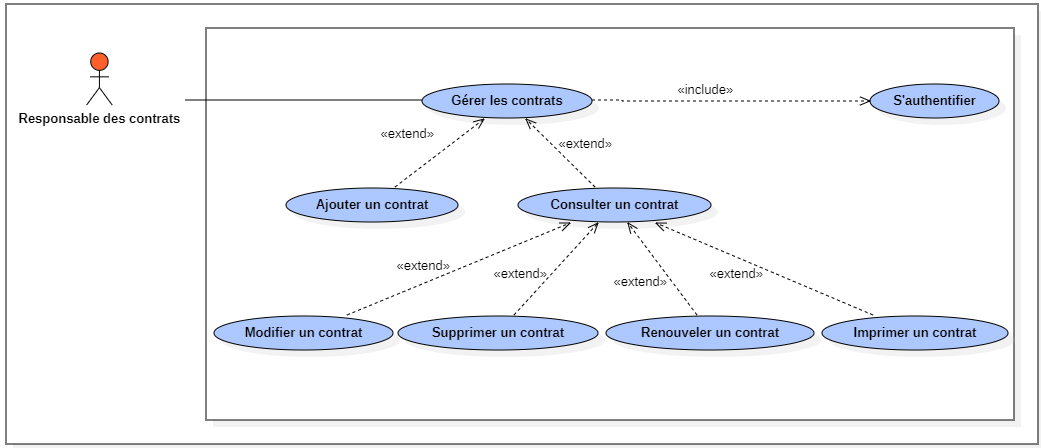
****

Figure 5: Gérer les contrats

**Description textuelle du cas d’utilisation « Gérer les contrats»**

|  |
| --- |
| Titre : Gérer les contrats client/fournisseur |
| Acteur : Le responsable des contrats |
| Précondition : S’authentifier |
| Post condition : Mise à jour des contrats |
| Les cas possibles :  \*Ajouter un contrat  \*Consulter un contrat  \*Modifier un contrat  \*Supprimer un contrat  \*Renouveler un contrat  \*Imprimer un contrat |

**Tableau 3 : Description textuelle du cas d’utilisation « Gérer les contrats»**

**Conclusion**

Tout au long de ce chapitre, nous avons essayé de donner une idée générale sur le projet à réaliser à travers une étude théorique et une étude de l'existant. Par conséquent, nous avons opté qu'avant l'implémentation du nouveau système il faut analyser et spécifier les différents besoins qui fera objectif du chapitre suivant.

**Chapitre3 : Conception**

**Introduction**

Le processus de conception est une étape très importante, elle nous permet de définir la vue qui décrite tous les aspects d'un système avec le concept objet. En d'autre terme cette étape nous permet de spécifier les fonctionnalités du futur système d'information.

On va utiliser le langage UML pour présenter la conception.

1. **Architecture de l’application**
   1. **Architecture MVC**

Le MVC signifie Model-View-Controller est une architecture de développement qui vise à séparer une application en trois composants logiques principaux : modèle, vue et le contrôleur.

Chacun de ces composants est construit pour gérer des aspects de développement spécifiques d’une application.

MVC est l’une des architectures les plus fréquemment utilisés dans l’industrie pour créer des projets extensibles et évolutifs.

Dans l'architecture MVC, les trois composants sont les suivants :

* modèle : données (accès et mise à jour)
* vue : interface utilisateur (entrées et sorties)
* contrôleur : gestion des événements et synchronisation

**Rôle de la vue :**

Une vue est un moyen d’afficher des objets dans une application. Par exemple, l’affichage d’une fenêtre ou des boutons ou d’un texte dans une fenêtre. Il comprend tout ce que l’utilisateur peut voir.

La vue est l’interface utilisateur. La vue permet à l’utilisateur d’afficher les données à l’aide d’un modèle et lui permet également de modifier les données.

**Rôle du modèle :**

Un modèle est le noyau de l’application, il gère les données, permet de récupérer les informations dans la base de données, de les organiser pour qu’elles puissent ensuite être traitées par le contrôleur.

**Rôle du contrôleur :**

Le contrôleur agit comme une interface entre le modèle et la vue, pour traiter toute le logique métier et les requêtes entrantes, manipuler les données à l’aide du composant Modèle et interagir avec les Vues pour rendre le résultat final.

* Les trois parties du MVC sont interconnectées.
* La vue affiche le modèle pourl’utilisateur.
* Le contrôleur accepte les entrées de l’utilisateur et met à jour le modèle et la vue

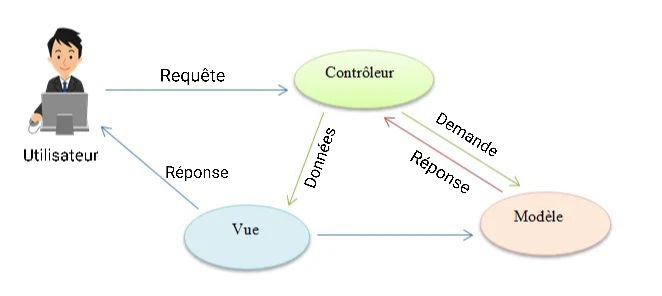
****

Figure 6 : Architecture en MVC

1. **Les vues de la conception :**
   1. **Vue statique : Le diagramme de classe**

Notre diagramme de clase contient 8 classes qui sont :

1. **La table utilisateur :**

|  |  |
| --- | --- |
| Attribut | Description |
| Id\_utilisateur | L’identifiant de l’utilisateur |
| Nom | Le nom de l’utilisateur |
| Prénom | Le prénom de l’utilisateur |
| Adresse | L’adresse de l’utilisateur |
| Ville | La ville de l’utilisateur |
| Mail | L’adresse mail de l’utilisateur |
| Num\_tel | Le numéro du téléphone de l’utilisateur |

**Tableau 4 : Description de la table utilisateur**

1. **La table client :**

|  |  |
| --- | --- |
| Attribut | Description |
| Id\_client | L’identifiant du client |
| Nom | Le nom du client |
| Prénom | Le prénom du client |
| Adresse | L’adresse du client |
| Ville | La ville du client |
| Mail | L’adresse mail du client |
| Num\_tel | Le numéro du téléphone du client |

**Tableau 5 : Description de la table client**

1. **La table fournisseur :**

|  |  |
| --- | --- |
| Attribut | Description |
| Id\_fournisseur | L’identifiant du fournisseur |
| Nom | Le nom du fournisseur |
| Prénom | Le prénom du fournisseur |
| Adresse | L’adresse du fournisseur |
| Ville | La ville du fournisseur |
| Mail | L’adresse mail du fournisseur |
| Num\_tel | Le numéro du téléphone du fournisseur |

**Tableau 6 : Description de la table fournisseur**

1. **La table administrateur :**

L’administrateur hérite tous les attributs de la table utilisateur sauf l’identifiant car chaque utilisateur a un identifiant unique.

|  |  |
| --- | --- |
| Attribut | Description |
| Id\_administrateur | L’identifiant de l’administrateur |

**Tableau 7 : Description de la table administrateur**

1. **La table Responsable des contrats client :**

Le Responsable des contrats clients pilote les ventes et contribue à la bonne gestion des contrats client élaborées au sein du CNI.

|  |  |
| --- | --- |
| Attribut | Description |
| Id\_responsable\_contat\_client | L’identifiant du responsable des contrats clients |

**Tableau 8 : Description de la table Responsable des contrats client**

1. **La table Responsable des contrats fournisseurs :**

Le Responsable des contrats fournisseurs pilote les achats des matières premières et composants nécessaires à la production.

|  |  |
| --- | --- |
| Attribut | Description |
| Id\_responsable\_contrat\_fournisseur | L’identifiant du responsable des contrats |

**Tableau 9 : Description de la table Responsable des contrats fournisseurs**

1. **La table des contrats client :**

Elle continue toutes les informations relatives au contrat d’un client.

|  |  |
| --- | --- |
| Attribut | Description |
| Réf\_contrat | La référence du contrat |
| Réf\_marché | La référence du marché |
| Id\_client | L’identifiant du client |
| Objet | L’objet du contrat |
| Montant | Le montant du contrat |
| Durée | La durée du contrat |
| Date\_début | La date de début du contrat |
| Date\_fin | La date de la fin du contrat |
| Mise\_en\_demeure | La mise en demeure du contrat |
| Structure\_technique | La structure technique du contrat |
| Piéce\_jointe | Le document d’accompagnement du contrat |
| Ajouter () | La fonction d’ajout du contrat |
| Modifier () | La fonction de la modification du contrat |
| Supprimer () | La fonction de la suppression du contrat |
| Consulter () | La fonction de la consultation du contrat |
| Renouveler () | La fonction du renouvellement du contrat |

**Tableau 10 : Description de la table contrat d’un client**

1. **La table des contrats fournisseurs :**

Elle continue toutes les informations relatives au contrat d’un fournisseur.

|  |  |
| --- | --- |
| Attribut | Description |
| Réf\_contrat | La référence du contrat |
| Réf\_marché | La référence du marché |
| Id\_fournisseur | L’identifiant du fournisseur |
| Objet | L’objet du contrat |
| Montant | Le montant du contrat |
| Durée | La durée du contrat |
| Date\_début | La date de début du contrat |
| Date\_fin | La date de la fin du contrat |
| Mise\_en\_demeure | La mise en demeure du contrat |
| Structure\_technique | La structure technique du contrat |
| Piéce\_jointe | Le document d’accompagnement du contrat |
| Ajouter () | La fonction d’ajout du contrat |
| Modifier () | La fonction de la modification du contrat |
| Supprimer () | La fonction de la suppression du contrat |
| Consulter () | La fonction de la consultation du contrat |
| Renouveler () | La fonction du renouvellement du contrat |

**Tableau 11 : Description de la table contrat d’un fournisseur**

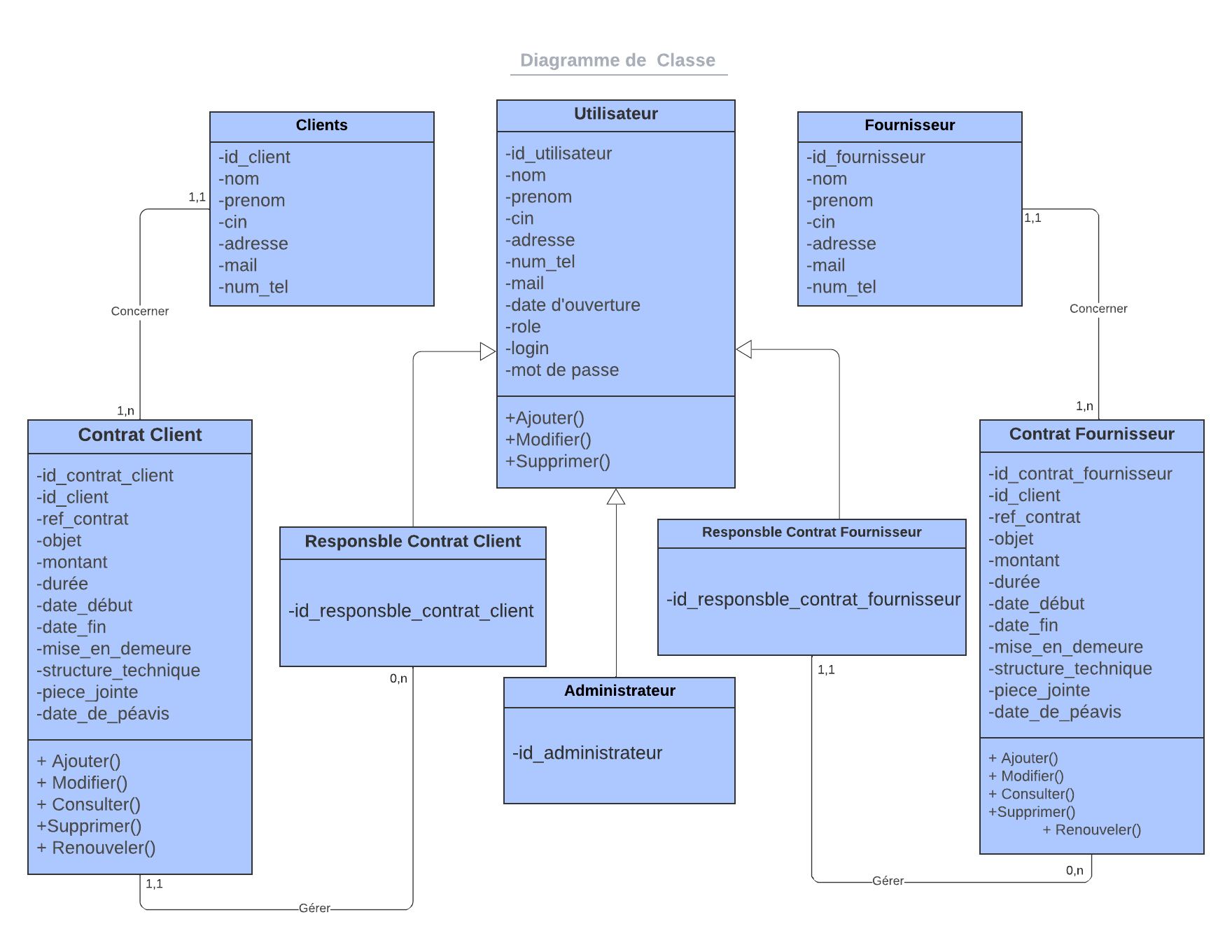


Figure 7: Le diagramme de classe

* 1. Vue dynamique : Les diagrammes de séquences

Le diagramme de séquence est la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

On montre ces interactions dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Le but étant de décrire comment se déroule les actions entre les acteurs ou objets.

* 1. Diagramme de séquence « S’authentifier»

Description textuelle du digramme de séquence « S’authentifier»

|  |
| --- |
| Titre : S’authentifier |
| Objectif : Se connecter pour utiliser les fonctionnalités de l’application |
| Acteur : L’utilisateur du système |
| Pré condition :L'utilisateur doit disposer un compte préalablement créé par l'administrateur système. |
| *Post condition* : Le système autorise ou non l'accès à la page demandée à l'utilisateur. |
| Scénario principal :  \*L’utilisateur demande l’accès au système  \*Le système affiche l’interface du login  \*L'utilisateur saisit son nom d'utilisateur, son mot de passe et son mail.  \* Le système vérifie le nom d’utilisateur, son mot de passe et son mail.  \*Le système affiche l'espace approprié pour chaque utilisateur. |
| Scénario alternatif :  Si les paramètres de connexion sont erronés, le système affiche un message d’erreur, on effectue un retour vers la page d'authentification. |

**Tableau 12 : Description textuelle du diagramme de séquence « S’authentifier »**

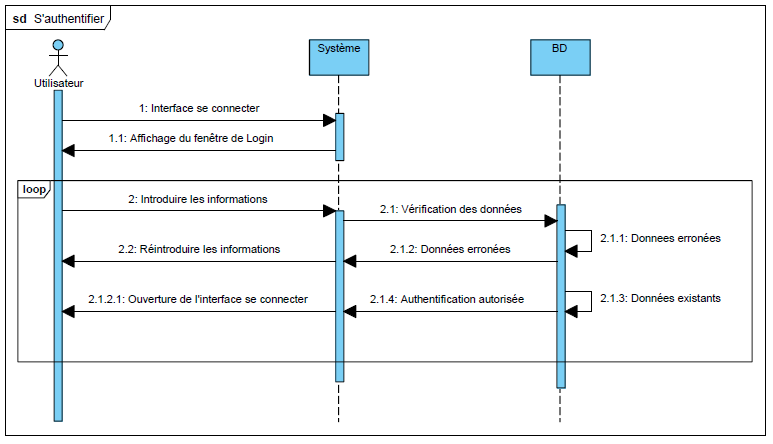


Figure 8 : S'authentifier

* 1. Diagramme de séquence « Modifier un utilisateur  »

Description textuelle du digramme de séquence « Modifier un utilisateur »

|  |
| --- |
| Titre : Modifier un utilisateur |
| Objectif : permettre à l'administrateur de modifier un compte utilisateur. |
| Acteur : L’administrateur |
| Pré condition : l'administrateur doit être sur la page d'accueil de l'application |
| Post condition de succès: le compte a été modifié  Post condition d'échec: le compte n'a pas été modifié |
| Scenario principal :  \* l'administrateur choisit l'utilisateur à modifier.  \*Le système affiche les informations relatives au compte.  \*l'administrateur modifie les champs voulus.  \*Le système met à jour les informations dans la base de données. |
| Scénario alternatif :  \*l'administrateur invalide la modification |

**Tableau 13 : Description textuelle du diagramme de séquence « Modifier un utilisateur »**

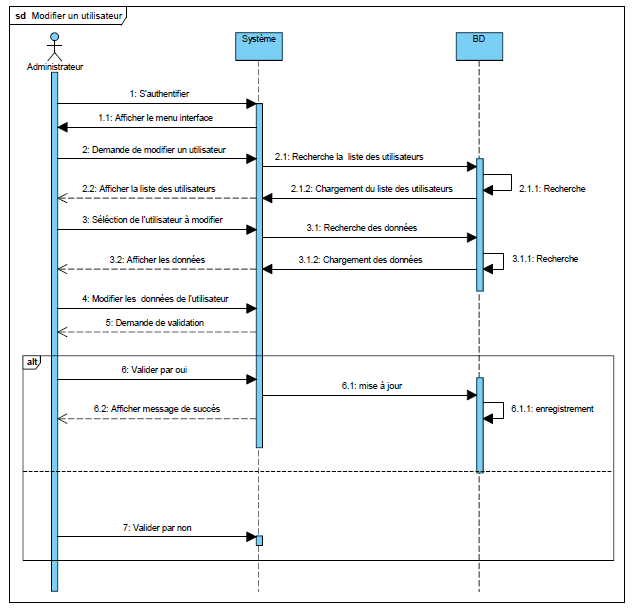


Figure 9 : Modifier un utilisateur

* 1. Diagramme de séquence « Ajouter un contrat»

Description textuelle du digramme de séquence « Ajouter un contrat»

|  |
| --- |
| Titre : Ajouter un contrat |
| Objectif : permettre au responsable des contrats d’ajouter un contrat. |
| Acteur : Le responsable des contrats |
| Pré condition : le responsable des contrats doit être sur la page d'accueil de l'application |
| Post condition de succès: le contrat a été crée  Post condition d'échec: le contrat n'a pas été crée |
| Scenario principal :  \*Le responsable des contrats doit s’authentifier  \* Le système affiche le menu principal de l’interface  \*Le responsable clique sur ajouter un contrat  \* Le système affiche le formulaire de création  \* le responsable des contrats saisie ses informations et valide  \* Le système vérifie les informations saisies et valide  \* Le système enregistre les informations  \* Le système envoie le message de succès |
| Scénario alternatif :  \*Le système vérifie et ne valide pas le formulaire  \*Retour au menu principal de l’interface |

**Tableau 14 : Description textuelle du diagramme de séquence « Ajouter un contrat »**

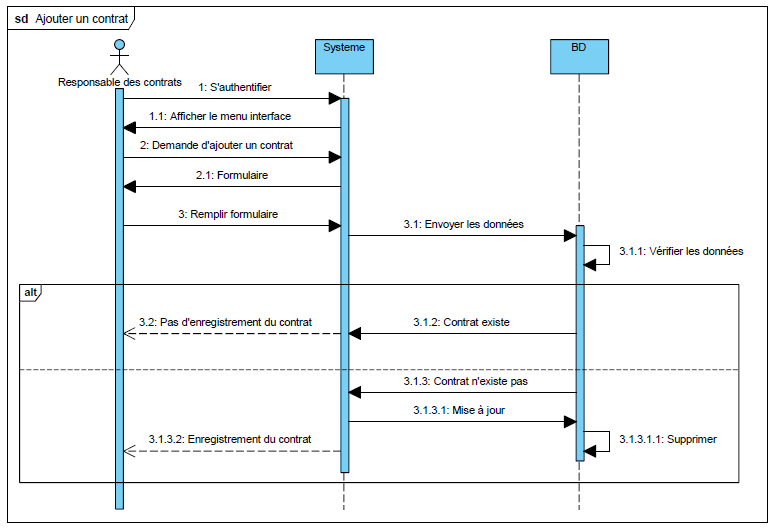


Figure 10 : Ajouter un contrat

* 1. Diagramme de séquence « Supprimer un contrat »

Description textuelle du digramme de séquence « Supprimer un contrat»

|  |
| --- |
| Titre : Supprimer un contrat |
| Objectif : permettre au responsable des contrats de supprimer un contrat. |
| Acteur : Le responsable des contrats |
| Pré condition : le responsable des contrats doit être sur la page d'accueil de l'application |
| Post condition de succès: le contrat a été supprimé  Post condition d'échec: le contrat n'a pas été supprimé |
| Scenario principal :  \*Le responsable des contrats doit s’authentifier  \* Le système affiche le menu principal de l’interface  \*Le responsable clique sur supprimer un contrat  \*Le système affiche le message de confirmation  \*Le responsable des contrats valide la suppression  \*Le système supprime le contrat et notifie  \* Le système enregistre la suppression |
| Scénario alternatif  \* Le responsable des contrats invalide la suppression  \* Le système réaffiche la liste des comptes |

**Tableau 15 : Description textuelle du diagramme de séquence « Supprimer un contrat »**

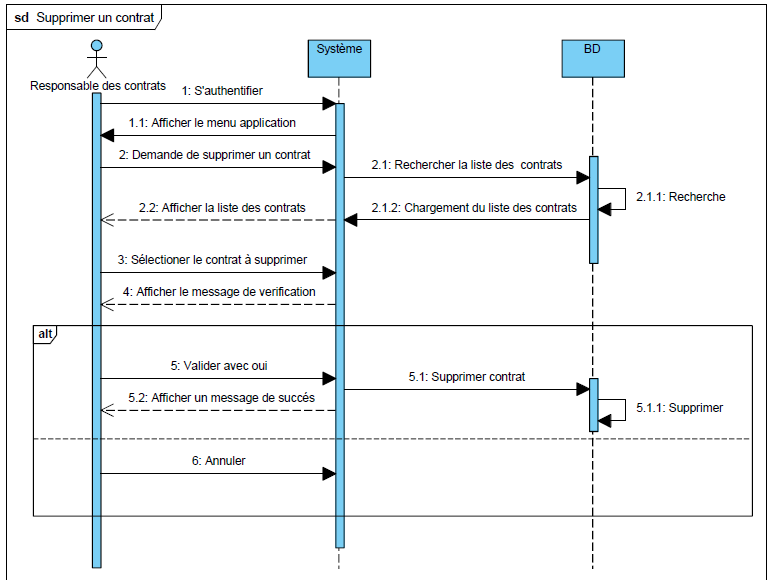


Figure 11 : Supprimer un contrat

2.7. Diagramme de séquence « Consulter un contrat »

Description textuelle du digramme de séquence « Consulter un contrat»

|  |
| --- |
| Titre : Consulter un contrat |
| Objectif : permettre au responsable des contrats de consulter un contrat. |
| Acteur : Le responsable des contrats |
| Pré condition : le responsable des contrats doit être sur la page d'accueil de l'application |
| Scenario principal :  \*Le responsable des contrats doit s’authentifier  \* Le système affiche le menu principal de l’interface  \*Le responsable clique sur consulter un contrat  \*Le système affiche la liste des contrats dans la base des données.  \* Le responsable des contrats sélectionne le contrat à consulter.  \* Le système affiche les informations relatives du contrat. |

**Tableau 16 : Description textuelle du diagramme de séquence « Consulter un contrat »**

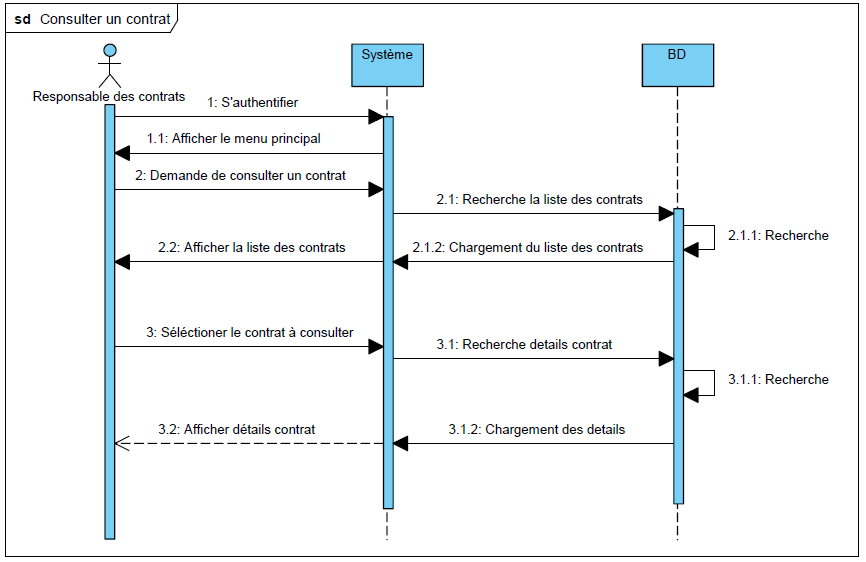


Figure 12 : Consulter un contrat

**Conclusion**

|  |  |
| --- | --- |
| A l'issue de ce chapitre nous avons appris à mieux connaitre la conception du système, c'est une phase très long et pénible, chaque étape de la conception est très importante, tout en respectant leurs normes grâce à la conception UML et le processus du développement en cascade, elle doit être effectuée en tenant compte des résultats des étapes qui la précèdent, elle permet de dégager l'architecture générale de notre application web représentée dans le prochain chapitre. |  |

**Chapitre 4 :**

**Développement et réalisation**

**Introduction**

La phase de réalisation est la plus importante étape dans le cycle de vie de notre application car à la fin de cette phase nous allons avoir le fruit de toutes les étapes précédentes. Pour réaliser notre application nous avons été face à plusieurs outils de développement.

Au cours de ce chapitre, nous allons présenter l'environnement matériel et logiciel ainsi que les différents choix techniques utilisés pendant cette phase. Puis, nous allons présenter les différentes interfaces de notre application.

**1. Environnement du travail**

**1.1 Environnement du travail matériel**

L'implémentation de cette application a été effectuée sur une machine dont la configuration matérielle est la suivante :

**Caractéristiques de la machine utilisée :**

- Modèle : HP

- Processeur : intel core i5

- Mémoire vive : 6Go

- Système d'exploitation : 64 bits

- Disque dur : 1To

-Cartes graphiques : AMD Radeon(TM) R5 M330 et intel(R) HD Graphics 5500

**1.2 Environnement du travail logiciel**

Les logiciels de développement utilisé dans ce stage sont :

* **Visual studio code**

Visual Studio Code est un éditeur de code simplifié, qui est gratuit et développé en open source par Microsoft. Il fonctionne sous Windows, mac OS et Linux. Il fournit aux développeurs à la fois un environnement de développement intégré avec des outils permettant de faire avancer les projets techniques, de l’édition, à la construction, jusqu’au débogage.

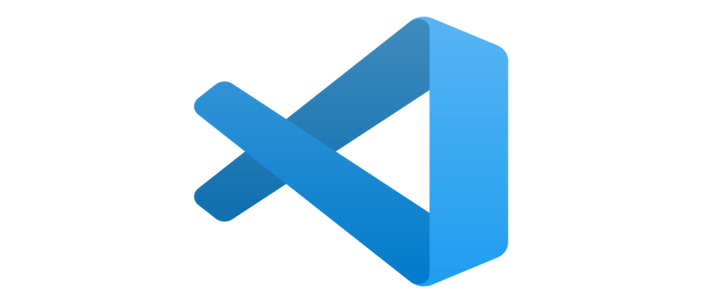
****

Figure 13 : logo de Visual studio code

* **XAMPP**

**XAMPP** est un ensemble de logiciels permettant de mettre en place facilement un serveur Web confidentiel, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (**X** (cross) **A**pache **M**ySQL **P**erl **P**HP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes

****

Figure 14 : logo de XAMPP

* **Notepad++**

Notepad++, un éditeur de texte gratuit et open source qui vous permet non seulement de taper du texte au format txt, mais surtout de créer des lignes de code avec prise en charge de la coloration des balises en fonction du langage que vous utilisez .

Il permet aux développeurs et aux utilisateurs souhaitant créer du code informatique de le faire de façon simple et intuitive. Son interface est simple et légère

****

Figure 15 : logo de Notepad++

**1.2.1. Les langages de programmation**

* **UML**

UML (Unified Modeling Language), est un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

Il permet de modéliser l'ensemble des données et des traitements en élaborant des différents diagrammes.



Figure 16 : Logo d’UML

* **HTML**

HTML est un acronyme de HyperText Markup Language, désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web. Il permet, entre autres, d'écrire de l'hypertexte, mais aussi d'introduire des ressources multimédias dans un contenu.



Figure 17 : Logo de HTML

* **CSS**

CSS est l’acronyme de « Cascading Style Sheets » ce qui signifie feuille de style en cascade.

Le CSS correspond à un langage informatique permettant de mettre en forme des pages web ([HTML](https://atinternet.com/glossaire/html) ou XML).Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les [couleurs](https://www.futura-sciences.com/sciences/definitions/physique-couleur-4126/), les marges et espacements, les bordures, les images de fond, etc.



Figure 18 : Logo de CSS

* **Node.js**

Node.js est un environnement JavaScript multiplateforme open source pour le développement d'applications côté serveur et de mise en réseau.

Node.js fournit également une riche bibliothèque de divers modules JavaScript qui simplifie dans une large mesure le développement d'applications Web utilisant Node.js.



Figure 19 : Logo de Node.js

**1.2.2. Les Frameworks**

* **Angular**

Angular est un framework de développement, construit sur Type Script. En tant que plateforme, Angular comprend :

Un cadre basé sur des composants pour la création d'applications web évolutives.

Une collection de bibliothèques bien intégrées qui couvrent une grande variété de fonctionnalités.

Une suite d'outils de développement pour vous aider à développer, construire, tester et mettre à jour votre code.

****

Figure 20 : Logo d'angular

* **Bootstrap**

Bootstrap est un [framework](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/) développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages [HTML](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/), CSS et JavaScript fournit aux développeurs des outils utiles à la création du design (graphisme, animation) de sites et d’applications web. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de [framework](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/) un "Front-End Framework".

****

Figure 21 : Logo de Bootstrap

**2. Interface graphique**

**2.1. Interface « Authentification »**

La page d’authentification permettre aux utilisateurs de se connecter en entrant le login et le mot de passe pour accéder au système.

L’utilisateur doit spécifier s’il est l’administrateur ou bien le responsable des contrats client ou fournisseur.

Si les informations saisies ne sont pas valides un message d’erreur s’affiche.

**2.2. Interface « «administrateur »**

Si l’utilisateur est un administrateur, la liste des comptes des utilisateurs du système s’affiche.

L’administrateur doit gérer les comptes en ajoutant, modifiant ou supprimant un compte d’un utilisateur.

L’administrateur peut aussi consulter la liste des contrats client ou bien fournisseur.

**2.2.1. Ajouter un compte utilisateur**

Pour ajouter un nouveau compte, l’administrateur doit cliquer sur l’interface « ajouter un compte », un formulaire sera affiché, il doit alors le remplir puis le valider en cliquant sur le bouton « enregistrer ».

**2.2.2. Modifier un compte utilisateur**

Pour modifier un compte, l’administrateur doit cliquer sur le bouton « Modifier », les détails du comptes seront affiché, il doit alors modifier les champs voulus puis valider les modifications apportées à ce compte en cliquant sur le bouton « enregistrer ».

**2.2.3. Supprimer un compte utilisateur**

Pour supprimer un compte, l’administrateur doit cliquer sur le bouton « Supprimer », un message de confirmation sera affiché, il clique alors sur le bouton « supprimer» pour valider la suppression du compte.

**2.3. Interface « Responsable des contrats »**

Dans notre application, on a deux types des responsables des contrats :

-Responsable des contrats client : Celui qui est responsable de la gestion des contrats de la vente.

- Responsable des contrats fournisseur : Celui qui est responsable de la gestion des contrats d’achat.

Si l’utilisateur est un responsable des contrats client/fournisseur, la liste des contrats client/fournisseur sera affichée (selon le droit d’accès).

Le responsable des contrats client/fournisseur peut consulter la liste des contrats client/fournisseur.

Le responsable des contrats client/fournisseur doit gérer les contrats en ajoutant, modifiant, renouvelant ou supprimant un compte d’un utilisateur.

**2.3.1. Ajouter un contrat**

Pour ajouter un contrat de type client ou bien fournisseur, le responsable des contrats client/fournisseur clique sur l’interface « Ajouter un contrat », un formulaire s’affiche, il doit alors le remplir puis cocher le bouton adéquat pour spécifier s’il s’agit d’un contrat de type client ou bien fournisseur, enfin il doit enregistrer ces informations en cliquant sur le bouton« Enregistrer ».

**2.3.2. Modifier un contrat**

Pour modifier un contrat de type client ou bien fournisseur, le responsable des contrats client/fournisseur clique sur l’interface « Modifier un contrat », les détails du contrat seront affiché, il doit alors modifier les champs voulus puis valider les modifications apportées au contrat en cliquant sur le bouton « enregistrer ».

**2.3.3. Supprimer un contrat**

Pour supprimer un contrat de type client ou bien fournisseur, le responsable des contrats client/fournisseur clique sur l’interface « supprimer un contrat », un message de confirmation sera affiché, il clique alors sur le bouton « supprimer» pour valider la suppression du contrat.

**2.3.4. Renouveler un contrat**

Pour renouveler un contrat de type client ou bien fournisseur, le responsable des contrats client/fournisseur clique sur l’interface « renouveler un contrat », les détails relatives au contrat seront affichés, il doit alors saisir la nouvelle date du fin puis la valider en cliquant sur le bouton « «enregistrer ».

**2.3.5. Consulter un contrat**

Cette interface permet au responsable des contrats client/fournisseur ou bien à l’administrateur de consulter la liste des contrats et d’afficher les informations relatives de ces derniers.

Ils peuvent sélectionner le contrat à consulter en cliquant sur le bouton « Consulter ».

**2.3.6. Imprimer un contrat**

Cette interface permet au responsable des contrats et à l’administrateur d’imprimer un contrat en cliquant sur le bouton « imprimer ».

**2.3.7. Rechercher un contrat**

Cette interface permet au responsable des contrats de rechercher un contrat selon le numéro de référence.

**Conclusion**

Ce chapitre offre une description plus ou moins détaillée des interfaces que nous avons développées. Il étale un peu l'aspect réalisation de notre application. Toutefois, notre travail reste ouvert pour des extensions et des améliorations éventuelles.

***Conclusion Générale***

Notre stage a été une expérience enrichissante qui nous a permis d’une part d’appliquer et d’approfondir nos connaissances acquises tout au long de notre formation et d’autre part, de nous familiariser avec la conduite des projets informatiques et d’explorer le milieu professionnel.

Le travail réalisé dans le cadre de notre projet de fin d’études consiste à développer une application web performante pour le centre national informatique qui permet essentiellement de gérer suivre les contrats clients et fournisseurs

Ce travail est décomposé en quatre étapes. :

La première a été consacrée à comprendre le contexte général du projet. Au cours de cette étape, nous avons essayé de structurer et définir les besoins attendus du futur système. Nous avons réussi à dégager les lacunes du système actuel et suggérer des solutions afin d'y remédier.

La deuxième étape a été dévolue à la spécification des besoins fonctionnels et non fonctionnels ce qui a permis de classer les fonctionnalités du système selon la priorité aussi à distinguer les différents acteurs interagissant avec l’application visée.

La troisième étape était la conception, dans laquelle nous avons fixé la  
structure globale de l’application. C’était une étape importante dans la représentation des différentes fonctionnalités de l’application

La dernière étape de notre projet était la partie réalisation qui a été consacrée à la présentation  
des outils du travail et les interfaces les plus significatives de notre application.

La réalisation de ce projet, nous a permis de créer un système de gestion et suivi des contrats.

A titre général, ce stage a constitué une expérience professionnelle exceptionnelle. La  
vie universitaire ne permet que d'avoir un aperçu de son futur métier. On découvre rapidement  
que l'aspect pratique dépasse souvent les connaissances acquises et qu'il constitue une  
formation rapide.

**Webographie**

**(1)** [**https://www.memoireonline.com/**](https://www.memoireonline.com/)**.**

**(2)** [**https://www.google.com/search?client=firefox-b-d&q=w3school**](https://www.google.com/search?client=firefox-b-d&q=w3school)**.**

**(3)** [**https://openclassrooms.com/fr/**](https://openclassrooms.com/fr/)**.**

**(4)** [**https://visionarymarketing.com/fr/2018/01/excel-pire-ennemi-entreprises/**](https://visionarymarketing.com/fr/2018/01/excel-pire-ennemi-entreprises/)**.**

**(5)** [**https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/**](https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/)**.**

**(6)** [**https://waytolearnx.com/2020/01/quest-ce-que-le-modele-mvc-et-a-quoi-ca-sert.html**](https://waytolearnx.com/2020/01/quest-ce-que-le-modele-mvc-et-a-quoi-ca-sert.html)**.**

**(7)** [**https://www.journaldunet.fr/web-tech/**](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/) **.**

**(8)** [**https://www.futura-sciences.com/tech/**](https://www.futura-sciences.com/tech/definitions/internet-css-4050/) **.**

**(9)** [**https://www.blogdumoderateur.com/tools/**](https://www.blogdumoderateur.com/tools/)**.**