Table des matières

[Introduction Générale 41](#_Toc394939740)

[CHAPITRE I Présentation et Analyse du Problème 43](#_Toc394939741)

[1. Présentation de l’Entreprise : 44](#_Toc394939742)

[PGH en bref ... 44](#_Toc394939743)

[2. Elaboration du Cahier des Charges : 45](#_Toc394939744)

[2.1- Etude de l’Existant 45](#_Toc394939745)

[2.1.2- Objectifs 45](#_Toc394939746)

[2.2- Présentation du Projet 45](#_Toc394939747)

[2.3- Glossaire 46](#_Toc394939748)

[2.3.1- Environnement de Développement 46](#_Toc394939749)

[2.3.2- Environnement d’Exécution 46](#_Toc394939750)

[2.4- Planning 46](#_Toc394939751)

[3. Méthodologie de Travail : 47](#_Toc394939752)

[3.1- Définition de cycle de vie 47](#_Toc394939753)

[3.2- Approche Itérative et Incrémentale 47](#_Toc394939754)

[4. Conclusion : 48](#_Toc394939755)

[CHAPITRE II Spécification des Besoins 49](#_Toc394939756)

[1. Analyse des Besoins : 50](#_Toc394939757)

[1.1- Besoins non Fonctionnels 50](#_Toc394939758)

[1.2- Besoins Fonctionnels 50](#_Toc394939759)

[1.3- Identification des Acteurs 51](#_Toc394939760)

[2. Diagramme de Cas d’Utilisation : 52](#_Toc394939761)

[3. Fiches Descriptives des Cas d’Utilisation : 53](#_Toc394939762)

[4. Conclusion : 54](#_Toc394939763)

[CHAPITRE III Conception de la Solution 55](#_Toc394939764)

[1. Méthode de Conception : 56](#_Toc394939765)

[1.1- Diagrammes de Séquence 56](#_Toc394939766)

[1.2- Diagramme de Classes 56](#_Toc394939767)

[2. Conception : 57](#_Toc394939768)

[2.1- Présentation des Diagrammes de Séquence 57](#_Toc394939769)

[2.2- Présentation de Diagramme de Classes 61](#_Toc394939770)

[3. Conclusion : 61](#_Toc394939771)

[CHAPITRE IV Mise en Œuvre et Validation 62](#_Toc394939772)

[1. Environnement de Travail : 63](#_Toc394939773)

[1.1- Configuration Matérielle 63](#_Toc394939774)

[1.2- Configuration Logicielle 63](#_Toc394939775)

[2. Architecture de l’Application : 63](#_Toc394939776)

[3. Imprimes-Ecran : 64](#_Toc394939777)

[Conclusion 70](#_Toc394939778)

[Bibliographie 71](#_Toc394939779)

Liste des figures

[Figure 1 : Calendrier des tâches 45](#_Toc394939811)

[Figure 2 : Schéma explicatif du principe 46](#_Toc394939812)

[Figure 3 : Schéma explicatif des processus 47](#_Toc394939813)

[Figure 4 : Diagramme de cas d’utilisation 52](#_Toc394939814)

[Figure 5 : Authentification 57](#_Toc394939815)

[Figure 6 : Ajout d’une actualité 58](#_Toc394939816)

[Figure 7 : Consultation des applications 59](#_Toc394939817)

[Figure 8 : Diagramme de classes 60](#_Toc394939818)

[Figure 10 : Architecture de la solution 63](#_Toc394939819)

[Figure 11 : Serveur SQL 64](#_Toc394939820)

[Figure 12 : Structure linQ 64](#_Toc394939821)

[Figure 13 : Interface d’ajout d’une application 65](#_Toc394939822)

[Figure 14 : Interface d’affectation des applications aux filiales 65](#_Toc394939823)

[Figure 15 : Interface d’ajout d’un service Web 66](#_Toc394939824)

[Figure 16 : Interface de consultation des services Web 66](#_Toc394939825)

[Figure 17 : Interface de consultation des applications 67](#_Toc394939826)

[Figure 18 : Interface de recherche des applications par filière 67](#_Toc394939827)

[Figure 19 : Interface d’envoi des réclamations 68](#_Toc394939828)

#### Introduction Générale

Vivre dans un environnement où les technologies sont révolutionnaires est un élément stimulant la tendance des entreprises à concurrencer l’avancement dans ce monde informatique.

L’entreprise, qui ne se contente plus d’utiliser les ordinateurs pour seulement gérer les documents d’une manière électronique et accéder à l’Internet, aspire à suivre l’évolution dans son entourage en soyant utilisateur des applications. Cette cause affecte aussi bien les besoins de la clientèle que les objectifs des entreprises.

C’est dans ce cadre qu’on peut classer le motif d’existence de ce sujet, objet d’un stage. En effet, le projet s’articule sur le développement d’un prototype d’application web permettant d’administrer un portail Sharepoint. Cette idée se dessine dans l’ensemble des « e-business solutions », auxquelles font appel les missions du département « Développement Informatique » de l’entreprise d’accueil « Poulina Group Holding » : association entre employés, partenaires et clients pour une information permanente et accessible sur des différentes plateformes.

A l’image de ce projet, le travail présenté dans ce rapport est structuré (en plus de l’introduction) en quatre chapitres :

Le premier chapitre est consacré à l’analyse et présentation du problème. Une introduction de l’entreprise d’accueil a précédé la formulation du besoin à travers un cahier des charges fixant les critères et les niveaux correspondants des fonctions de service. Le chapitre est terminé par l’énoncé de la méthodologie de travail en se fondant sur la définition du cycle de vie du projet.

Le deuxième chapitre fait appel aux différents critères (fonctionnels et non fonctionnels) liant le produit (application) à l’acteur (utilisateur et administrateur) à l’aide des fiches descriptives du diagramme de cas d’utilisation.

En optant pour le langage UML, la conception de la solution est exposée en détails dans le troisième chapitre traitant les diagrammes de séquence et le digramme de classes relatifs à la modélisation du problème étudié.

Le quatrième chapitre est destiné à la mise en œuvre du résultat de la conception : l’environnement de travail et l’architecture de la solution (données, interfaces et graphiques, design et styles) sont détaillés pour témoigner enfin la validation de la solution au moyen des figures (imprimes-écran de l’application).

Le rapport se termine par une conclusion qui intègre des perspectives pour le programme de travail à venir.

CHAPITRE I   
**Présentation et Analyse du Problème**

Dans ce chapitre, on commence par vous présenter l’entreprise Poulina Group Holding ainsi qu’une étude qui va vous donner un aperçu général sur tous les processus de développement et de gestion des projets. Le but est de répondre aux besoins du client et de découvrir les nouvelles technologies. L’utilité de ce chapitre est donc de faire ressortir le cadre du projet.

# Présentation de l’Entreprise :

## PGH en bref ...

Le groupe Poulina a été constitué en 1967 à l'initiative de promoteurs tunisiens privés dans le secteur avicole. Le dynamisme de l'entreprise articulé autour d'une intégration stratégique verticale et horizontale a permis de développer un groupe multisectoriel oeuvrant dans des activités diverses (l'agriculture, l'industrie, l'agroalimentaire et le service).

En 41 ans, le groupe Poulina est devenu le premier groupe à capitaux privés du pays, il compte actuellement plus de 70 sociétés dont une quinzaine établies à l’étranger. Le groupe possède des implantations en Algérie, Maroc, la Lybie et récemment en Arabie Saoudite et en Chine.

Dans le cadre de la restructuration menée en prélude à son introduction en bourse, Poulina vient de créer en juin 2008, une holding Baptisée Poulina Group Holding (PGH) composée de six mini-holdings :

* Poulina : opérant principalement dans les secteurs d’alimentation, d’élevage et

d’abattage.

* Agro-services (Poulina Holding SICAF – PHS): opérant principalement dans le secteur agroalimentaire et de service.
* Industrie et Technique : opérant dans le secteur Industriel.
* Carthago : opérant dans le secteur de fabrication et commercialisation de céramique.
* Unipack : opérant dans le secteur de l’emballage.
* Ettaamir : opérant dans le secteur de la promotion immobilière.

# Elaboration du Cahier des Charges :

## 2.1- Etude de l’Existant

### 2.1.1- Problématique

La mise à jour du portail exploité au sein de l’entreprise est compliquée et lourde puisqu’elle est réalisée seulement en accédant à la base de données via un gestionnaire pour apporter physiquement les changements. Ce qui cause le gaspillage du temps ainsi que la nécessité de l’intervention d’un informaticien pour accomplir cette tâche.

## 2.1.2- Objectifs

* Eviter la circulation inutile des données (exemple : entre le rédacteur des actualités et l’informaticien) ainsi que le risque des mauvaises manipulations.
* Assurer la disponibilité maintenue des informations et des nouveautés.
* Diminuer le taux d’accès manuel à la base de données.
* Garantir la facilité de la gestion en assemblant dans un seul processus tous les services de mise à jour du portail et de son administration.
* Affirmer la simplicité d’usage des outils informatiques par des non-informaticiens.
* Centraliser l’information.
* Prendre en considération les cas d’exception afin de les résoudre.
* Mettre en évidence le respect des rôles et des fonctions.

## 2.2- Présentation du Projet

Ce travail réside à mettre en œuvre une solution informatique consistant en un prototype d’application Web qui permettra d’administrer le portail Sharepoint exploité au sein de PGH.

Elle met à la disposition de l’administrateur une interface de gestion des utilisateurs et de leurs droits d’accès…

En fait, elle introduit un véritable guide présentant  les services offerts (ajout, consultation, modification et suppression) sur les différents éléments des Web parts (actualités, applications, messagerie électronique, etc) qui constituent le portail.

## 2.3- Glossaire

## 2.3.1- Environnement de Développement

Pour pouvoir implémenter cette solution sous forme d’une application ayant comme langage de développement C Sharp (C#), on a choisi d’exploiter le Framework de programmation Web ASP.NET sous l’outil de développement Microsoft Visual Studio 2010 Proffessional.  
On va se servir également des services web décrits en eXtensible Markup Language (XML) et du Cascading Style Sheet (CSS) pour le design.

On va avoir recours ainsi à SQL Server 2005 afin de gérer la base de données.

## 2.3.2- Environnement d’Exécution

Cette application devrait être fonctionnelle sur tout appareil supportant le langage C# et doté d’une connexion de données.

## 2.4- Planning

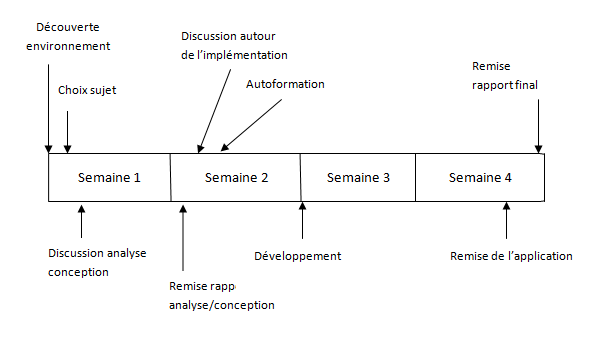


Figure 1 : Calendrier des tâches

# Méthodologie de Travail :

## 3.1- Définition de cycle de vie

Le cycle de vie d’un logiciel est l’ensemble de phases permettant sa construction et l’établissement des besoins du client jusqu’à son achèvement en tant que produit commercial.

Ce cycle de vie contient plusieurs activités dans différentes phases du développement du logiciel comme : la récolte des besoins, analyse du problème, conception, implémentation, etc.

## 3.2- Approche Itérative et Incrémentale

* *Itérative :* le processus de développement est appliqué plusieurs fois.
* *Incrémentale :* chaque itération augmente la quantité d’information.
* *Découpage du travail en mini-projets :* chaque mini-projet est une itération qui donne lieu à un incrément.
* *Chaque itération* doit finir par la réception des retours des utilisateurs.

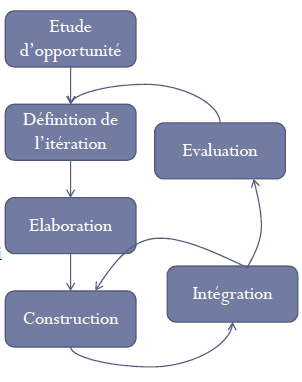


Figure 2 : Schéma explicatif du principe

* L’ordonnancement des itérations est basé sur les priorités entre cas d’utilisation et sur l’étude du risque.

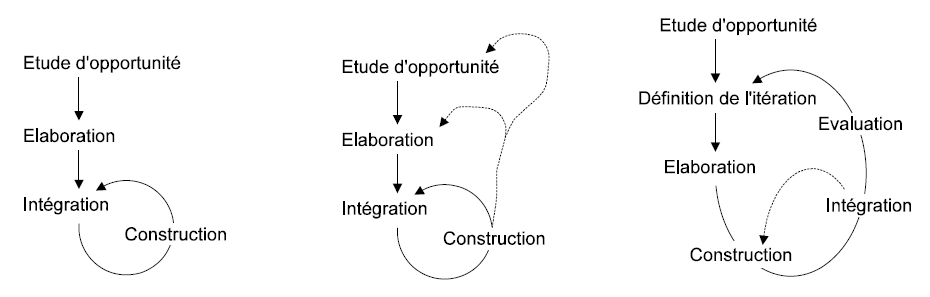


Figure 3 : Schéma explicatif des processus

* Segmentation du travail.
* Concentration sur les besoins et les risques.
* Les itérations sont des prototypes :
* Expérimentation et validation des technologies.
* Planification et évaluation.
* Les prototypes s’enroulent autour du noyau de l’architecture.

# Conclusion :

Ce chapitre consiste en une présentation générale de l’organisme d’accueil et une étude de l’existant qui ont permis de dévoiler les problèmes courants et de mettre en surface un cahier des charges qu’on va respecter.

CHAPITRE II   
**Spécification des Besoins**

Dans ce chapitre, on élaborera les spécifications relatives à cette application. On décrira dans un premier temps les spécifications non fonctionnelles et dans un second temps les spécifications fonctionnelles.

# Analyse des Besoins :

## Besoins non Fonctionnels

L’application doit être :

* *Ergonomique :* Assurer la discipline de l’adéquation entre l’utilisateur et l’application pour que cette dernière soit adaptée aux caractéristiques de l’Homme en employant des icones.
* *Conviviale :* Eliminer la complexité et diminuer le taux d’erreurs afin de faciliter l’utilisation de l’application en dirigeant l’utilisateur vers l’action qu’il faut faire et les données qu’il faut fournir à l’application (l’insertion du calendrier des dates, les listes déroulantes).
* *Informative :* Garantir la disponibilité persistante des données demandées par l’utilisateur.
* *Efficace :* Fournir les résultats les plus perforants qui répondent aux besoins de l’utilisateur.
* *Portable :* Cette application se caractérise par sa capacité de fonctionner dans différents environnements d’exécution sans exiger des contraintes matérielles spécifiques.
* *Cohérente :* Tous les éléments constituant l’application possèdent forcement un rapport d’organisation logique entre eux.
* *Fiable :* L’application doit exécuter correctement ; toute information qui lui est retournée doit êtres certaine (la crédibilité de la source des données).

## Besoins Fonctionnels

Cette application doit satisfaire les critères suivants :

* Gérer le contenu du portail :
* Tâches :
* Consultation.
* Gestion totale (consultation, ajout, modification, suppression).
* Messagerie :
* Consultation.
* Gestion totale (consultation, ajout, modification, suppression).
* Actualités :
* Gestion totale (consultation, ajout, modification, suppression).
* Applications :

- Catalogue :

* Consultation.
* Gestion totale (consultation, ajout, modification, suppression).

- Liste favoris :

* Gestion totale (consultation, ajout, modification, suppression).
* Liens rapides :
* Gestion totale (consultation, ajout, modification, suppression).
* Documents publiés :
* Gestion totale (consultation, ajout, modification, suppression).
* Surmonter des réclamations.
* Administrer les comptes utilisateurs :
* Consultation.
* Gestion des droits d’accès.

## 1.3- Identification des Acteurs

Un acteur est une entité qui interagit avec l’application, il peut être un utilisateur humain, un système qui fournit des services ou un système qui utilise les services de l’application. Un acteur peut jouer plusieurs rôles et un rôle peut être joué par plusieurs acteurs.

Les différents acteurs qui agissent sur ce système sont les suivants :

Utilisateur de l’application : c’est l’acteur principal qui aura recours à l’application afin de gérer le contenu du portail Sharepoint et profiter des services offerts.

Administrateur : c’est l’acteur qui aura recours à l’application afin de gérer les comptes utilisateurs en leur affectant les droits d’accès.

IS : c’est l’acteur secondaire qui aura la tâche de gestion de l’authentification afin d’autoriser et reconnaitre les utilisateurs.

# Diagramme de Cas d’Utilisation :

L’ensemble des cas d’utilisation permet de représenter le fonctionnement du système vis-à-vis de l’utilisateur en décrivant son comportement dans son environnement extérieur.

Ce diagramme sert à structurer les besoins des utilisateurs, leurs exigences et les objectifs de l’application.

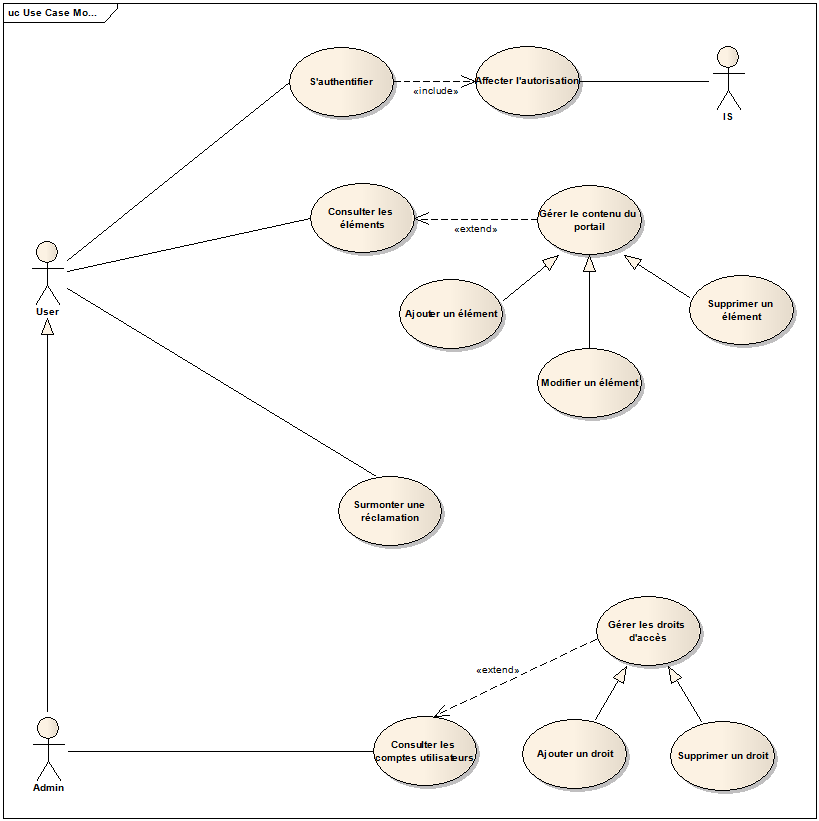


Figure 4 : Diagramme de cas d’utilisation

Ce diagramme de cas d’utilisation a permis d’extraire tous les besoins du client et ses différentes activités vis-à-vis le système, servant d’interfaces entre tous les intervenants de ce dernier.

# Fiches Descriptives des Cas d’Utilisation :

**Cas d’utilisation :** Consulter les éléments

**Acteur :** User

**Pré-condition :** Démarrage de l’application

**Scénario nominal :**

* + - * + Le système affiche la page d’accueil.
        + L’utilisateur choisit le service (actualités par exemple) en sélectionnant son icône dans le menu principal.
        + Le système envoie une requête au serveur Sharepoint.
        + Le système reçoit le résultat de la requête.
        + Le système formate les données importées puis les affiche.

**Scénario alternatif :** si l’utilisateur tente de réaliser une action de mise à jour (modification d’un élément) alors qu’il n’a pas droit de la faire, le système doit afficher une alerte.

**Cas d’utilisation :** Ajouter un élément

**Acteur :** User

**Pré-condition :** Démarrage de l’application et autorisation accordée

**Scénario nominal :**

* + - * + Le système affiche la page d’accueil.
        + L’utilisateur choisit le service (actualités par exemple) en sélectionnant son icône dans le menu principal.
        + Le système envoie une requête au serveur Sharepoint.
        + Le système reçoit le résultat de la requête.
        + Le système formate les données importées puis les affiche.
        + L’utilisateur clique sur le bouton d’ajout d’un élément.
        + Le système affiche un formulaire à remplir.
        + L’utilisateur fournit les données nécessaires.
        + Le système récupère ces données et il les envoie au serveur Sharepoint.
        + Le système met à jour la liste des éléments puis affiche le résultat.

**Scénario alternatif :** si l’utilisateur valide l’ajout et il existe encore des champs obligatoires du formulaire qui sont vides, le système doit afficher une alerte.

**Post-condition :** validation de l’ajout pour mettre à jour les informations.

**Cas d’utilisation :** Consulter les comptes utilisateurs

**Acteur :** Admin

**Pré-condition :** Démarrage de l’application

**Scénario nominal :**

* + - * + Le système affiche la page d’accueil.
        + L’administrateur choisit le service de consultation des utilisateurs en sélectionnant son icône dans le menu principal.
        + Le système envoie une requête au serveur Sharepoint.
        + Le système reçoit le résultat de la requête.
        + Le système formate les données importées puis les affiche.

**Scénario alternatif :** si l’administrateur tente de réaliser une action de mise à jour, le système doit afficher une alerte.

# Conclusion :

Dans ce chapitre, on a commencé par présenter les besoins auxquels vient répondre cette application pour passer ensuite à détailler les étapes de sa conception.

CHAPITRE III  
**Conception de la Solution**

Etant défini et analysé les différents besoins, on se focalise maintenant sur la phase de conception qui a pour objectif de permettre la formalisation des étapes préliminaires du développement d’un système dans le but de faciliter la réalisation et la maintenance.

# Méthode de Conception :

Le langage le plus apte, dans ce projet, est UML (Unified Modeling Language, qui se traduit par langage de modélisation unifié) pour la simple raison qu’il offre des moyens pour capturer les connaissances d’un sujet et les exprimer en vue de les communiquer. En plus, il dédie un ensemble de modèles qui facilitent la compréhension du problème.

Les caractéristiques de l’UML sont:

* Un langage sans ambiguïtés
* Un langage universel pouvant servir de support pour tout langage orienté objet
* Un moyen de définir la structure d’un programme
* Une représentation visuelle permettant la communication entre les acteurs d’un même projet.
* Une notation graphique simple, compréhensible même par des non informaticiens.
* UML permet de tirer profit des avantages de la conception orientée objet à savoir la facilité de la maintenance, l'évolutivité, la réutilisation des composants…

## Diagrammes de Séquence

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel. Ils représentent les échanges de message entre objets au cours du temps. Ils s’utilisent de deux manières: pour la documentation des cas d’utilisations (scénarios) et pour la représentation précise des interactions entre objets (usage plus informatique).

## Diagramme de Classes

C’est le diagramme pivot de la modélisation. Il exprime de manière générale la structure interne et statique d’un système, en termes de classes et de relations entre ces classes.

Une classe permet de décrire un ensemble d’objets (attributs et comportement), tandis qu’une relation ou association permet de faire apparaître des liens entre ces objets.

Nous pouvons donc distinguer :

* Un objet est une instance de classe
* Un lien est une instance de relation
* Un diagramme de classe est un modèle permettant de décrire de manière abstraite et générale les liens entre objets.

# Conception :

## 2.1- Présentation des Diagrammes de Séquence

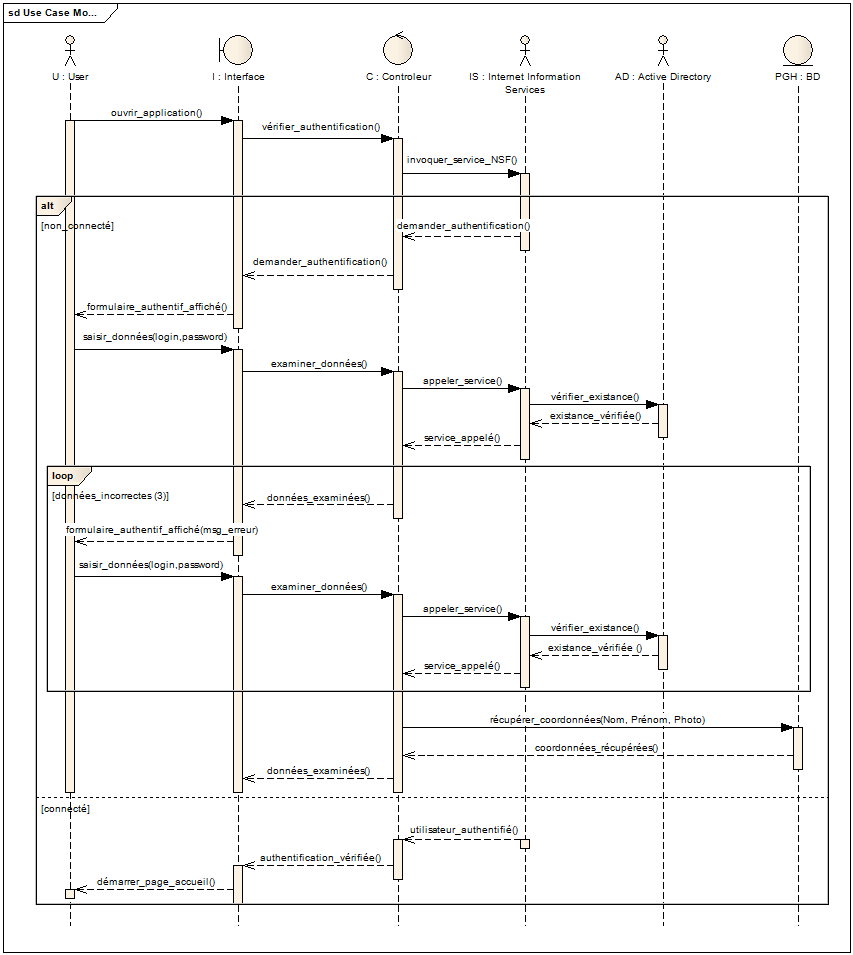


Figure 5 : Authentification

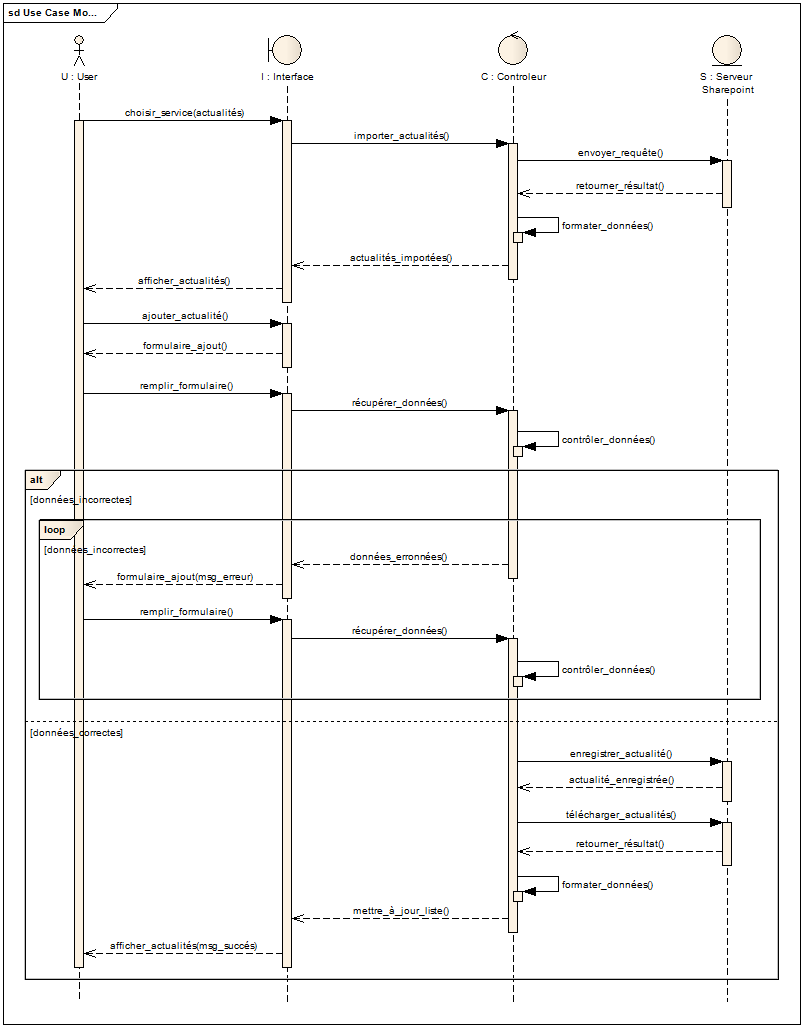


Figure 6 : Ajout d’une actualité

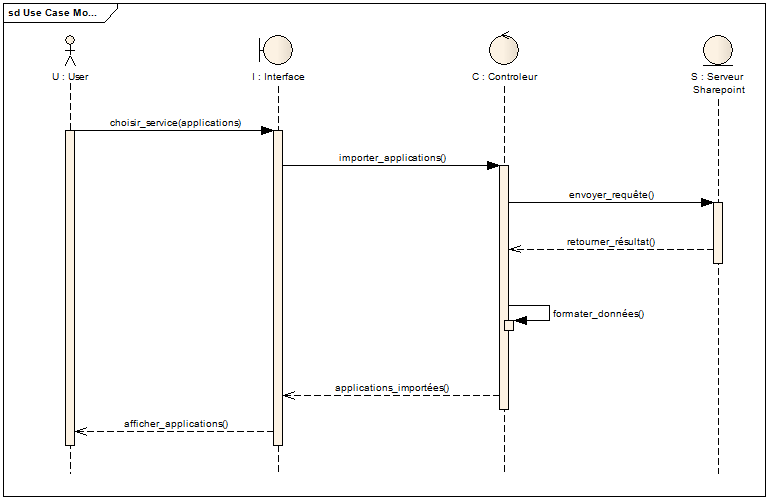


Figure 7 : Consultation des applications

Ces diagrammes de séquence ont aidés à déterminer la succession chronologique des opérations systèmes répondant aux activités réalisées par l’acteur (un utilisateur) pour atteindre l’objectif désiré.

## 2.2- Présentation de Diagramme de Classes

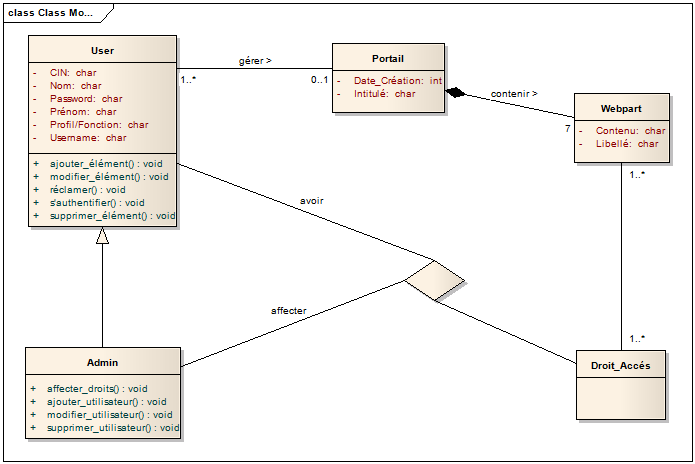


Figure 8 : Diagramme de classes

Ce diagramme de classes a permis de relever toutes les entités nécessaires pour le fonctionnement du système ainsi que leurs relations sémantiques s’appuyant sur l’intercommunication entre elles pour fournir un service.

# Conclusion :

Après avoir détaillé la conception de l’application en utilisant comme langage de modélisation le langage objet UML, on peut maintenant passer à la partie réalisation qui fera le sujet du chapitre suivant.

CHAPITRE IV  
**Mise en Œuvre et Validation**

Dans ce chapitre, on présentera la finalité du travail effectué durant ce stage. Pour cela on va initier ce thème par introduire l’environnement et l’ensemble des outils ayant contribués à sa réalisation.

# Environnement de Travail :

Pour la réalisation de ce projet, on a eu recours à un ordinateur portatif dont sa configuration est :

## Configuration Matérielle

* *Mohamed-Laptop :*
* Packard Bell.
* Processeur Intel® Core™ i5-4200U CPU @ 1.60GHz 2.30GHz.
* 10,0 Go de mémoire vive (RAM).
* 1 To d’espace disque.

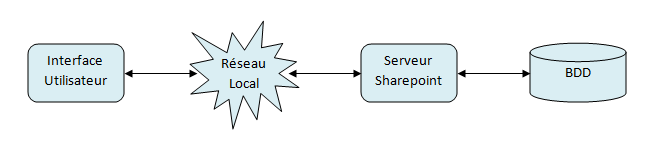
## Configuration Logicielle

* *Mohamed-Laptop :*
* Système d’exploitation : Microsoft Windows 7 Edition Intégrale.
* Microsoft Word 2007 pour la rédaction des documents.
* Enterprise Architect 7.1 pour la modélisation de la conception UML.
* Paint et Outil Capture pour la capture des imprimes-écran.

# Architecture de l’Application :

L’architecture de cette application peut être considérée comme étant une architecture Client/Serveur puisque dans la plupart des temps, l’application invoque des opérations décrites par des services Web afin de réaliser une tâche spécifique sans avoir besoin de connaître la manière avec laquelle s’exécute.

Cette application peut s’étaler en trois niveaux : données, traitement des données et présentation ou affichage des données.

**Figure 9 : Architecture de l’application**

# Imprimes-Ecran :

|  |
| --- |
| architecture.PNG |
| Figure 10 : Architecture de la solution |
| archi_bo.PNG |
| Figure 11 : Serveur SQL |

|  |
| --- |
| archi_dal.PNG |
| Figure 12 : Structure linQ |
| archi_gui_css_xml.PNG |
| Figure 13 : Interface d’ajout d’une application  affecterappFiliale.PNG |

Figure 14 : Interface d’affectation des applications aux filiales

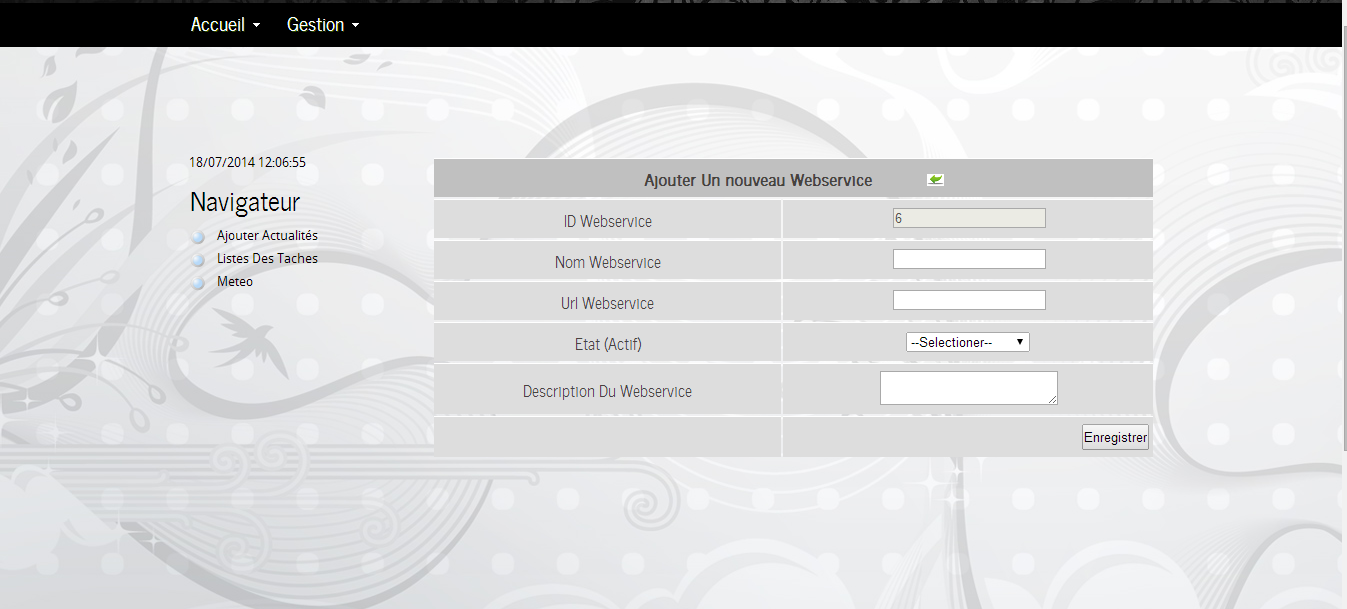


Figure 15 : Interface d’ajout d’un service Web



Figure 16 : Interface de consultation des services Web

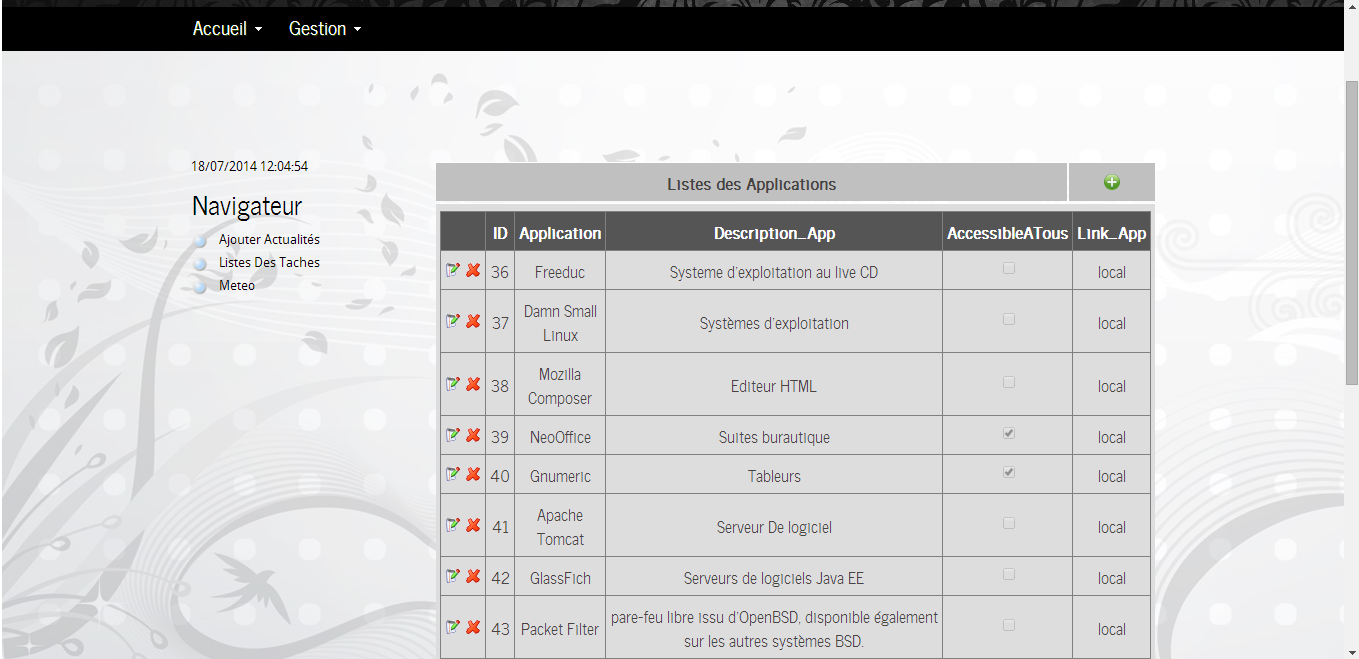


Figure 17 : Interface de consultation des applications

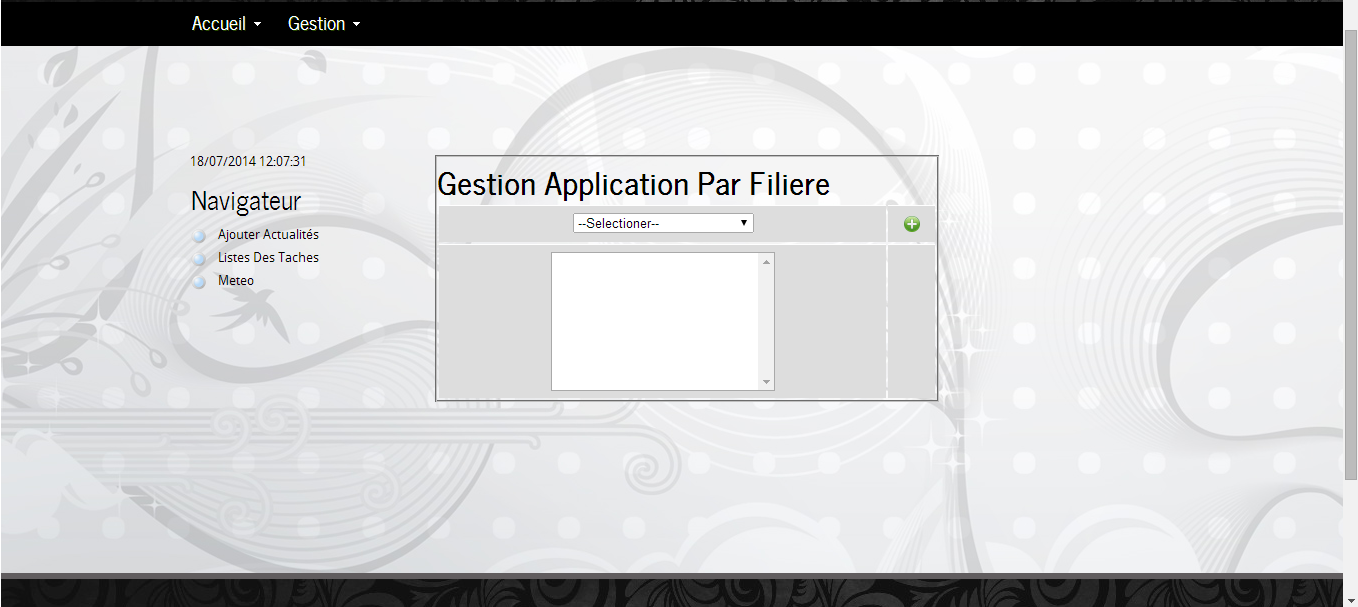


Figure 18 : Interface de recherche des applications par filière

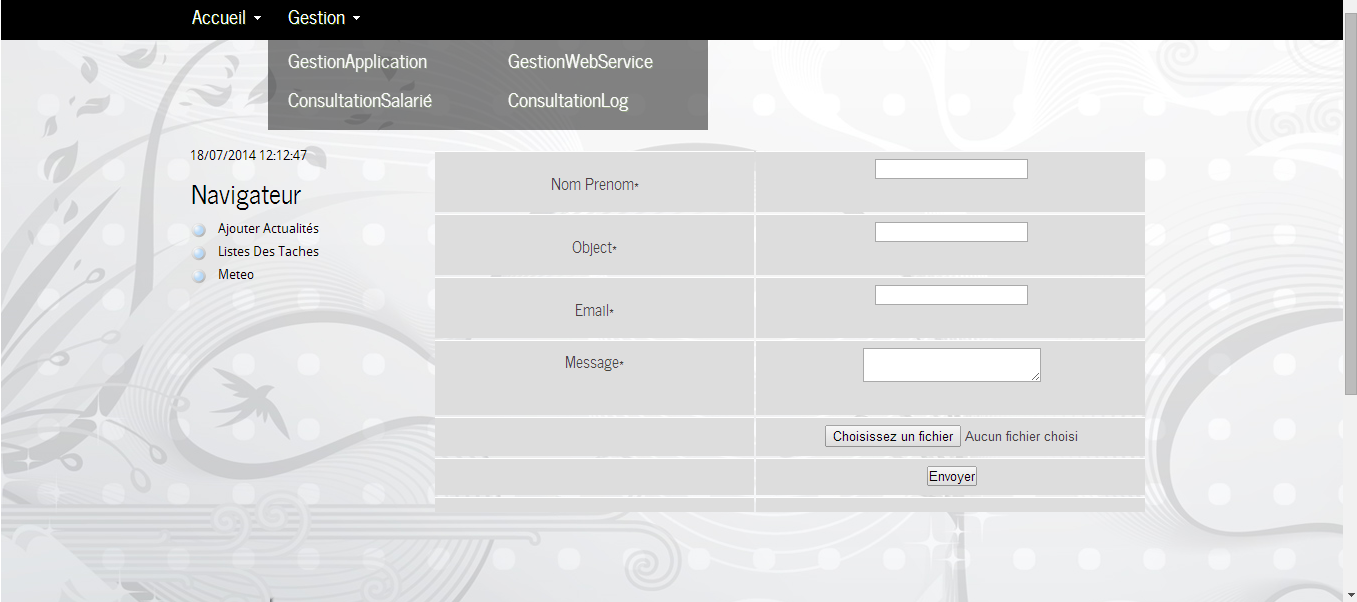


Figure 19 : Interface d’envoi des réclamations

#### Conclusion

On a cherché, à travers ce projet, à modéliser et concevoir une solution informatique dont l’exploitation est ciblée aux différents employés au sein de PGH : une interface d’administration et de gestion de contenu du portail Sharepoint. La réponse à ce besoin est sous forme d’application web.

Pour ce faire, on a entamé le travail par une formulation du cahier des charges. Le besoin a été explicité en soulignant les objectifs, appréciant la manière dont les fonctions de service sont remplies ainsi que leurs performances. Ayant choisi C Sharp, on s’est référé de suite au Framework ASP.NET sous Microsoft Visual Studio 2010 pour sa richesse et sa facilité d’usage. La phase d’énoncé des raisons et objectifs de production du projet est suivie par une conception de la solution. La dernière phase a ciblé la « réalisation du projet » : l’implémentation de l’application qui demande comme données à faire entrer le login et le password et affiche, après la sélection d’un service, comme résultat une interface d’administration du composant choisi. L’utilisateur a la possibilité de consulter ou mettre à jour le contenu du portail suivant ses droits d’accès.

Les effets de ce travail occasionnent une visée de répercussion basée sur un ensemble de prolongements :

* L’allocation des sessions serveurs pour chaque utilisateur dans le but de :
* L’autoriser à contribuer à l’amélioration de l’application et la fourniture des renseignements publiques.
* Lui affecter le droit de commenter l’application et de signaler les anomalies (bugs).

#### Bibliographie

[1] : Cours de conception, Introduction aux systèmes d’information et Modèles Orientés Objet, Mme Lilia SFAXI, 26/09/2010.

[2] : Cours de conception, Introduction à UML et Diagramme de Cas d’Utilisation, Mme Lilia SFAXI, 04/10/2010.

[3] : Cours de conception, Diagramme de Classes et Diagramme d’Objets, Mme Lilia SFAXI, 05/10/2010.

[4] : Cours de conception, Méthode de Conception Orientée Objet et le Langage UML, Mourali Olfa / Gallas Abir / Sfaxi Lilia / Zoubeir Najet / Turki Hazar / Krichi Med Khalil, 2010/2011.

[5] : Cours génie logiciel, le cycle de vie du logiciel, Luciana PROVENZANO,

<http://www-lagis.univ-lille1.fr/formations/sii/cours/Cours_CSI_6.pdf> , 2006.

[6] : Formation création des requêtes linQ, tutoriels,

http://msdn.microsoft.com/fr-fr/data/cc904318.aspx

[7] : Formation création des étapes d’un projet, tutoriels, http://projets.developpez.com/

[8] : Formation création des templates, tutoriels, fr.wix.com/website/templates