**Introduction :**

Nous présentons dans ce chapitre notre travail qui consiste à développer un site web pour l’échange d’information entre les étudiants du département informatique de la faculté de génie électrique et informatique.

Dans le but d’avoir une meilleure analyse et de rendre la conception de notre application plus complète, nous avons adopté le langage UML qui permet de bien représenter l’aspect statique et dynamique d’une application par une série de diagrammes qu’il offre.

**1. Les objectifs de notre application :**

L’objectif de notre application est de concevoir un site web pour l’échange d’information entre étudiants de notre département.

Notre application permettra :

**Au visiteur de :**

* Naviguer sur le portail du site
* Voir les différentes rubriques du site
* S’inscrire
* Voir les publications du forum

**A l’utilisateur de :**

* Accéder au forum
* Gérer les fichiers
* Gérer son profile
* Consulter les affichages
* Consulter les chroniques
* Voir les outils

**Et à l’administrateur de :**

* Gérer le contenu
* Gérer le forum
* Gérer les utilisateurs

**2. Présentation de l’UML :**

**2.1. Un bref historique d’UML :**

La technologie orienté objet est issue du monde de la programmation pour répondre à des besoins dans l’industrie du logiciel tel que la rapidité du développement, la réutilisation et la modularité. Ainsi des langages tels que SMALTALK, C++, ADA ont eu du succès dans ce domaine.

Dès lors, aux années 80 on commença à réfléchir à des langages de modélisation ou conception graphique. Les premiers ouvrages sur le sujet

parurent entre les années 1980 à1992, proposant des démarches ou méthodes de modélisation orientées objet, tels que :

**1) GRADY BOOCH :** Méthode **OOAD** « Object OrientedAnalysis and Design» en français « Analyse et Modélisation Orientée Objet ».

**2) PETER HOOD** : Méthode **OOD** « Object Oriented Design » en français

« Modélisation Orienté Objet ».

**3) IVAR JACOBSON** : Méthode **OOSE** « Oriented Object Software Engineering » en français « Génie Logiciel Orienté Objet ».

**4) JAMES RUMBAUCH** : Méthode **OMT** « Object ModelingTechnic » en français« Technique de Modélisation Objet ».

Tous les auteurs cités ci-dessus et bien d’autres proposaient des méthodes très proches les unes des autres puisque elles étaient toutes basées sur les concepts issus de l’orienté objet.

Cependant, chacun utilisait son propre formalisme ou notation. L’**OMG** (Object Management Group), posa, alors le problème de standardisation de ces formalismes. L’**OMG** est un consortium d’entreprises fondé pour construire des standards pour les systèmes orientés objet dans le but d’en assurer leur interopérabilité.

En 1995, **GRADY BOOCH** et **JAMES RUMBAUGH**, à l’époque chercheures chez Rational Software1, ont préparé une première description de leur méthode unifiée sous la version V.0.8. La même année, les auteurs sont rejoints par **IVAR JACOBSON** pour travailler sur le même projet.

En 1997, Rational propose la version 1.0 d’**UML** à l’**OMG**. Les auteurs cités sont considérés aujourd’hui comme les géniteurs d’**UML** et continuent à œuvrer pour son amélioration à travers ses différentes versions.

L’évolution des versions d’**UML** peut être récapitulée dans le tableau suivant :

|  |  |
| --- | --- |
| **Années** | **Versions** |
| 1995 | Méthode unifiée d’UML 0.8 (intégrant la méthode Booch) puis UML 0.9 (intégrant les méthodes OOSE et OMT). |
| 1996 | UML 1.0 proposé à l’OMG |
| 1997 | UML1.0 standardisé par l’OMG |
| 1998 | UML 1.2 |
| 1999 | UML 1.3 |
| 2000 | UML 1.4 |
| 2003 | UML 1.5 puis UML 2.0 |

Table II.1 : Évolution du langage UML

**2.2. Définition :**

UML est l’abréviation de UnifiedModelingLanguage, c'est-à-dire langage unifié pour la modélisation. C’est une notation graphique destinée à la création de modèles orientés objet en vue de l’analyse et de la modélisation de logiciels orientés objet. Ce n’est pas une méthode, c’est un ensemble d’outils permettant la modélisation de la future application informatique.

**2.3. Extension d’UML pour le web :**

L’extension d’UML pour le Web définit un ensemble de stéréotypes, d’étiquettes et de contraintes, qui rend possible la modélisation d’application Web. Ces stéréotypes et ces contraintes sont appliqués sur certains des composants propres aux applications Web, permettant ainsi de les représenter au sein du même modèle et sur les mêmes diagrammes que ceux qui décrivent le reste du système.

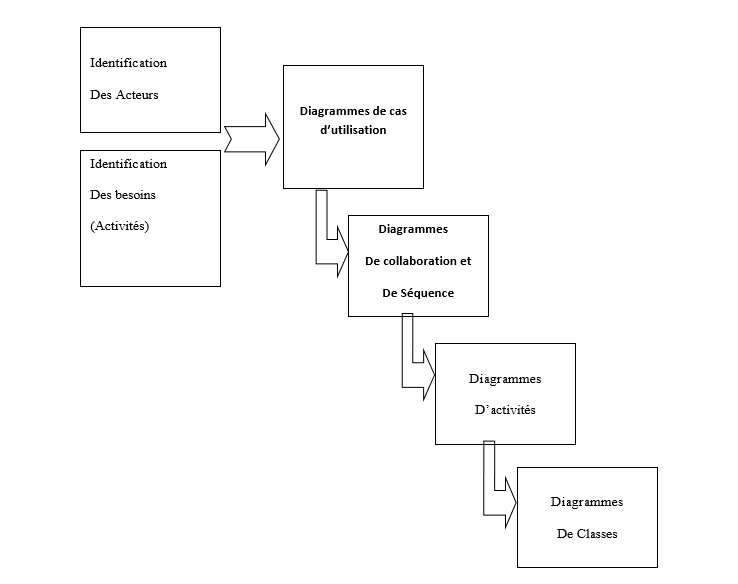
Le principal élément spécifique des applications Web étant la page Web , plusieurs stéréotypes qui lui sont destinés , sont décrits plus loin ainsi que d’autres , conçu pour les éléments tels que les cadres , les cibles et les formulaires représentant eux aussi des composants architecturales significatifs. Pour plus de détails voir l’annexe.

**2.3. Modélisation avec UML :**

UML permet de représenter des modèles, mais il ne définit pas les processus d’élaboration de ces modèles. Les auteurs d’UML conseillent tout de même une démarche pour favoriser la réussite d’un projet, cette démarche doit être :

* Une démarche itératif et incrémentale : Pour comprendre et représenter un système complexe, pour analyser par étapes, pour favoriser le prototypage et pour réduire et maitriser l’inconnu.
* Une démarche guidée par les besoins des utilisateurs : Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre à leur besoins. Chaque étape sera affinée et validée en fonction des besoins des utilisateurs.
* Une démarche centrée sur l’architecture logicielle : c’est la clé de voute du succès d’un développement, les choix stratégiques définiront la qualité du logiciel.

**2.4. La démarche de modélisation avec UML :**



Diagrammes De Classes

Diagrammes D’Activités

Diagrammes de Séquences

Identification Des Besoins (Activités)

Diagramme De Cas d’Utilisations

Identification Des Acteurs

Figure II.1 : Démarche de modélisation UML

**3. Analyse :**

Cette activité commence par l’étude des cas d’utilisations et de leurs scénarios ainsi que les besoins fonctionnels du système (ce que le système doit faire en réponse à une requête d’un utilisateur).

Ce modèle d`analyse est constitué par des classes et de collaboration des classes qui traduisent les comportements dynamiques et détaillés dans les cas d`utilisation et les besoins. Elle se concentre sur les besoins fonctionnels sans prendre compte des contraintes d`architectures de la machine ; pourvue que les besoins fonctionnels exprimés par les cas d`utilisations soient réalisables dans le système.

**3.1. Identifications des acteurs :**

Un acteur représente un ensemble cohérent de rôles joués par des entités externes (utilisateur humain, matériel ou autre système) qui interagissent directement avec le système étudié et pouvant consulter ou modifier directement l’état du système en émettant et/ou recevant des messages susceptibles d’être porteur de données.

**Les acteurs de notre système sont :**

* **le simple visiteur** : considéré comme une personne qui veut satisfaire sa curiosité.
* **l’utilisateur** : il bénéficie des différents services offerts par l’application.
* **l’administrateur** : personne qui a pour rôle principal de gérer toute l’application et qui a accès à tous.

**3.2. Diagramme de contexte :**

C’est un modèle conceptuel de flux qui permet d’avoir une vision globale des interactions entre le système et les liens avec l’environnement extérieur. Il permet aussi de bien délimiter le champ de l’étude.

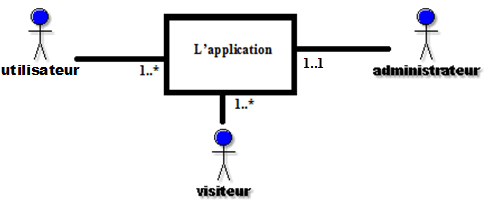
Pour notre cas le diagramme de contexte est le suivant :

Figure II.2 : Diagramme de contexte

**3.3. Spécification des taches :**

Les principales tâches assurées par les acteurs de notre système sont cités dans le tableau suivant :

|  |  |  |
| --- | --- | --- |
| **Acteurs** | **Taches** | **commentaire** |
| Visiteur | T0 : Naviguer sur le portail du site  T1 : Voir les différentes rubriques du site  T2 : S’inscrire  T3 : Accéder au forum |  |
| Utilisateur | T4 : S’authentifier  T5 : Participer au forum  T6 : Gérer les fichiers (cours/td/tp/emd)  T7 : Gérer son profile  T8 : Consulter les affichages  T9 : Consulter les chroniques  T10 : Voir les outils  T11 : Se déconnecter | T6 : gérer = ajouter rechercher  Consulter  signaler  T9 : chronique=des nouvelle du monde informatique |
| Administrateur | T12 : S’authentifier  T13 : Gérer les fichiers  T14 : Gérer le forum  T15 : Gérer les chroniques  T16 : Gérer les affichages  T17 : Gérer les outils  T18 : gérer les utilisateurs  T19 : Se déconnecter |  |

Table II.2 : Spécification des taches

**3.4. Spécification des scénarios :**

Un scénario est une suite d`échanges entre un ou plusieurs acteurs et le système, il permet de décrire un cas d`utilisation dans un contexte particulier. En d`autres termes, le scénario est un enchaînement ordonné et précis d`opérations pour réaliser un cas d`utilisation.

A chaque étape du scénario pourra se produire soit :

* un message de l`acteur vers le système.
* une validation ou un changement d`état du système.
* un message de système vers un acteur.

En décrivant les tâches définit auparavant par des scénarios, on aura le tableau suivant :

|  |  |  |
| --- | --- | --- |
| **Acteurs** | **Taches** | **Scénarios** |
| **Visiteur** | T0 | S0 : Sélectionner les liens  S1 : Accéder aux pages |
| T1 | S2 : voir les différentes rubriques du site |
| T2 | S3 : Sélectionner le lien <<inscription>>  S4 : Saisir le formulaire  S5 : Envoyer le formulaire |
| T3 | S6 : Sélectionner le lien <<forum>>  S7 : Voir les différents messages publiés |
| **Utilisateur** | T4 | S8 : Sélectionner le lien <<connexion>>  S9 : Saisir le pseudo et le mot de passe  S10 : valider |
| T5 | S11 : Sélectionner le lien << forum>>  S12 : Chercher un message  S13 : Consulter les messages  S14 : Écrire un message  S15 : Répondre à un message existant |
| T6 | S16 : Atteindre les fichiers  S17 : Rechercher  S18 : Consulter (voir/télécharger)  S19 : Ajouter  S20 : Signaler (un fichier indésirable) |
| T7 | S21 : Sélectionner le lien <<compte>>  S22 : Supprimer  S23 : Modifier |
| T8 | S24 : Sélectionner le lien <<affichage>>  S25 : Consulter les affichages  S26 : Recherche |
| T9 | S27 : Sélectionner le lien <<accueil>>  S28 : Consulter les chroniques  S29 : Rechercher |
| T10 | S30 : Sélectionner le lien <<outils>>  S31 : Rechercher  S32 : Suivre les liens pour télécharger |
| T11 | S33 : Sélectionner un lien<< déconnexion>> |
| **Administrateur** | T12 : idem que l’utilisateur | S8, S9, S10 |
| T13 | S16, S17, S18, S19  S34 : Supprimer des fichiers  S35 : Gérer le contenu signalé |
| T14 | S11, S12, S13, S14, S15  S36 : Supprimer des messages |
| T15 | S27, S28, S29  S37 : Ajouter  S38 : Supprimer |
| T16 | S24, S25, S26  S39 : Ajouter  S40 : Supprimer |
| T17 | S30, S31, S32  S41 : Ajouter  S42 : Supprimer |
| T18 | S43 : Sélectionner le lien <<utilisateur>>  S44 : Supprimer  S45 : Rechercher |
| T19 : idem que l’utilisateur | S33 |

Table II.3 : Spécification des scénarios

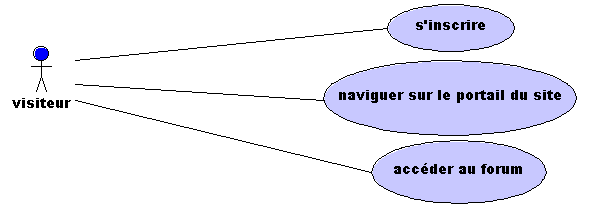
**3.5. Les cas d’utilisation :**

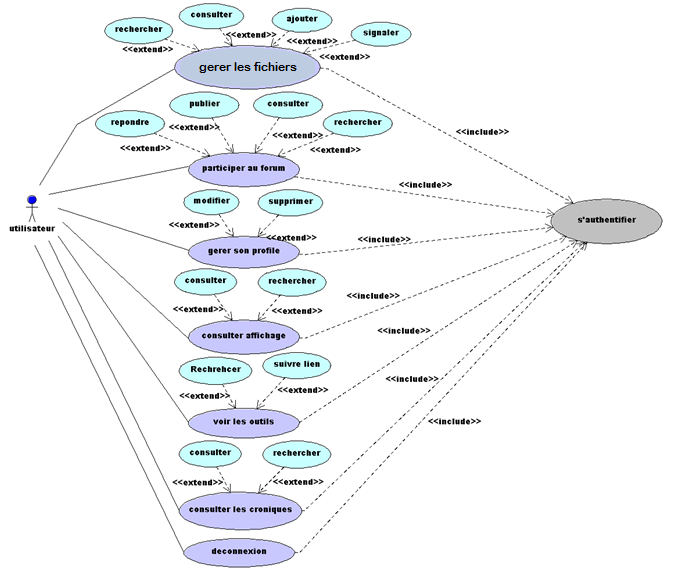
Tout système peut être décrit par un certain nombre de cas d’utilisation correspondant aux besoins exprimés par l’ensemble des utilisateurs.

Un cas d’utilisation représente un ensemble de séquences d’actions réalisées par le système qui produisent un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

**3.6. Diagramme de cas d’utilisation global :**

Un diagramme comportemental qui est une vue graphique de tous les acteurs d`un système, de ses cas d`utilisations et de leurs interactions.



****

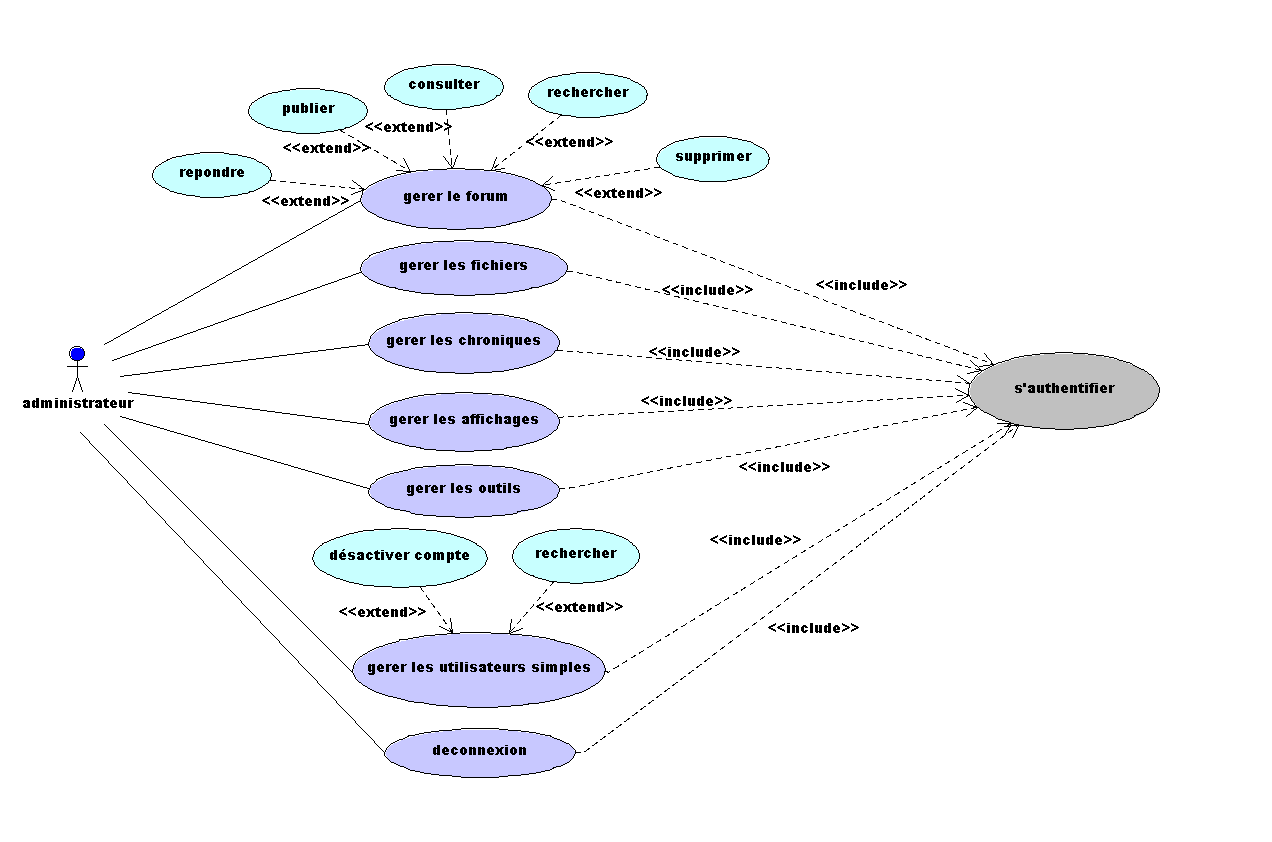
****

Figure II.3 : Diagramme de cas d’utilisation global

**4. Conception :**

Dans cette phase une nouvelle vue du modèle fait son apparition. Cette vue exprime les modules et les exécutables physiques sans aller à la réalisation concrète du système. On s’est basée sur :

* Les diagrammes de séquence
* Les diagrammes d'activités
* Les diagrammes de classe

Dans cette phase conception, nous représenterons quelques diagrammes de séquence, d’activité et de classes, correspondant aux cas d’utilisation déjà décrits.

**4.1. Les diagrammes de séquences :**

Ils présentent la vue dynamique du système. L’objectif du diagramme de séquence est de représenter les interactions entre les objets en indiquant la chronologie des échanges. Cette représentation se réalise par cas d’utilisation.

**4.1.1. Diagramme de séquence du cas d’utilisation « Authentification de l’administrateur» :**

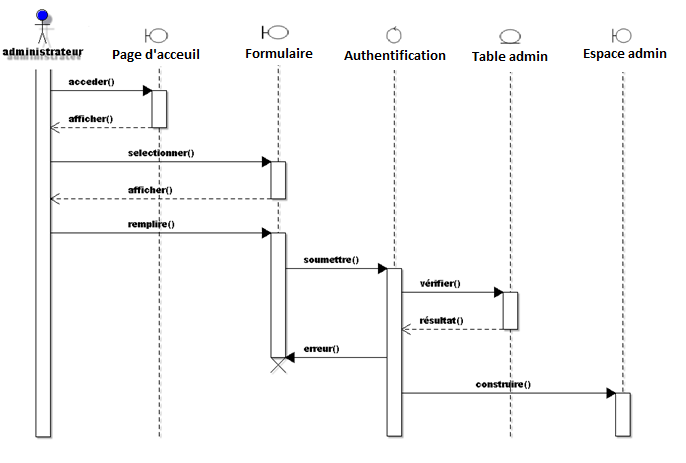
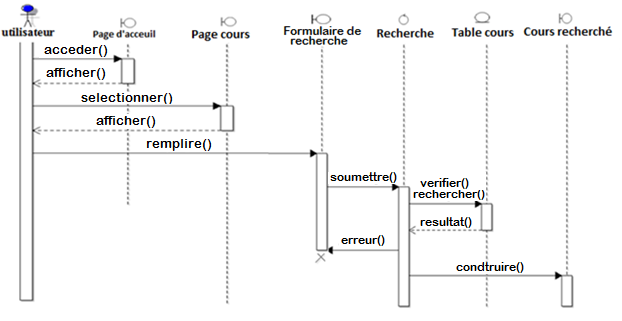
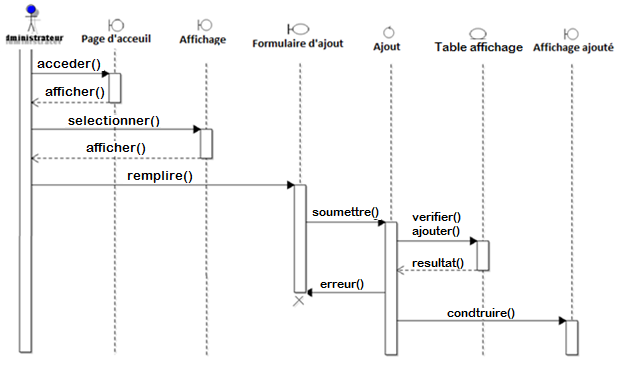
****

Figure II.4 : Diagramme de séquence ‘Authentification de l’administrateur‘

**4.1.2. Diagramme de séquence du cas d’utilisation « Recherche des cours par un utilisateur» :**

Figure II.5 : Diagramme de séquence ‘ Recherche des cours par un utilisateur ‘

**4.1.3. Diagramme de séquence du cas d’utilisation « Ajout d’un affichage par l’administrateur »**

Figure II.6 : Diagramme de séquence ‘ Ajout d’un affichage ‘

**4.2. Les diagrammes d’activités :**

Le diagramme d’activité est un organigramme utilisé pour modéliser l’aspect dynamique du système. Il permet de montrer le flot de control d’une application à une autre. Il permet aussi de modéliser les étapes séquentielles dans un processus de calcul.

**4.2.1. Diagramme d’activité du cas d’utilisation « Ajout d’un cours par un utilisateur» :**

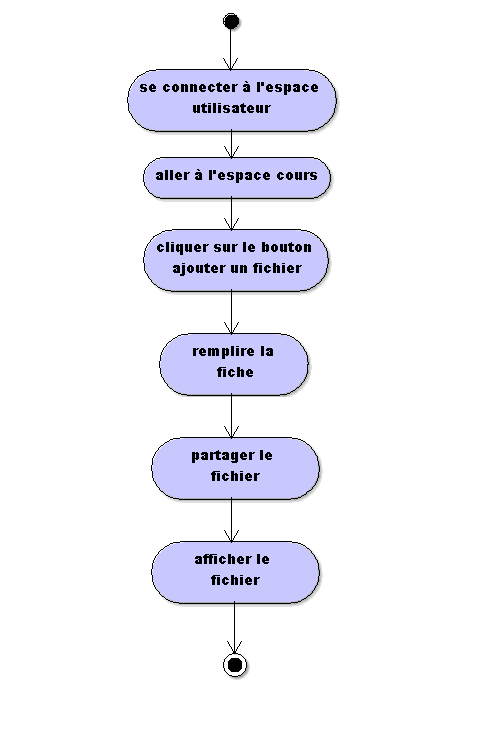
****

Figure II.7 : Diagramme d’activité« Ajout d’un cours par un utilisateur»

**4.2.2. Diagramme d’activité du cas d’utilisation « Suppression d’un utilisateur par l’administrateur» :**

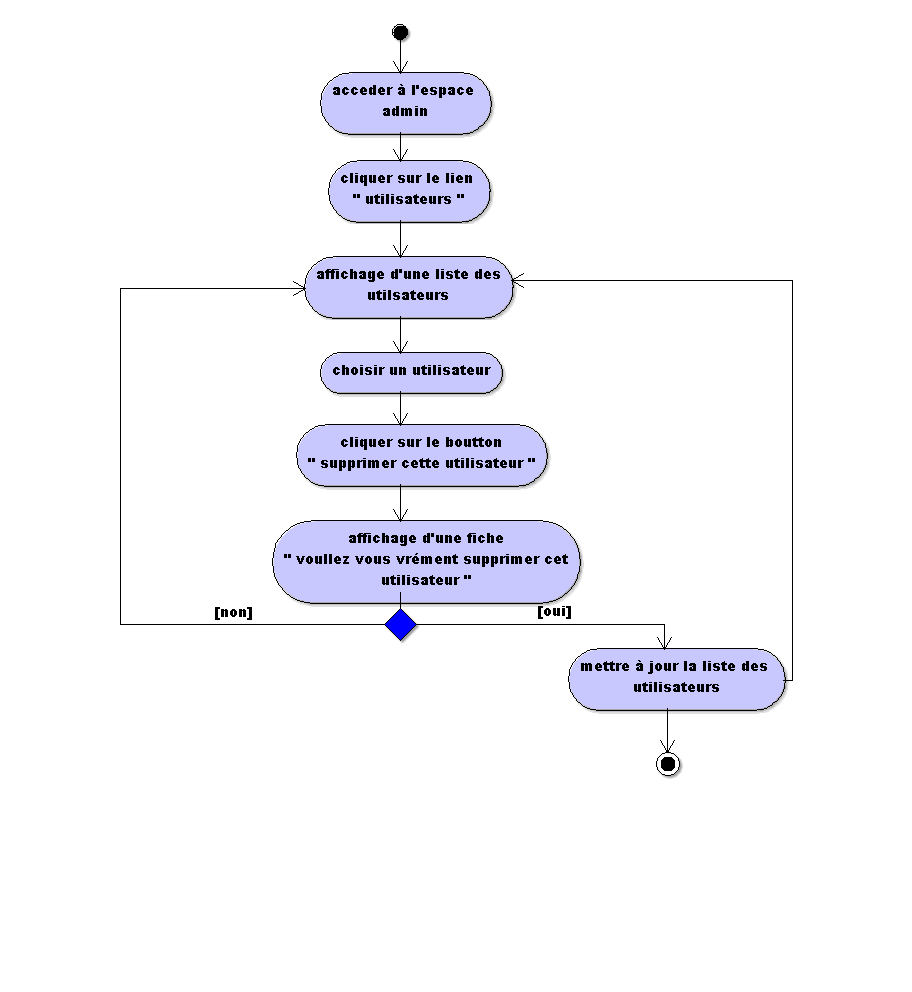
****

Figure II.8 : Diagramme d’activité «Suppression d’un utilisateur

« Par l’administrateur»

**4.2.3. Diagramme d’activité du cas d’utilisation « Inscription» :**

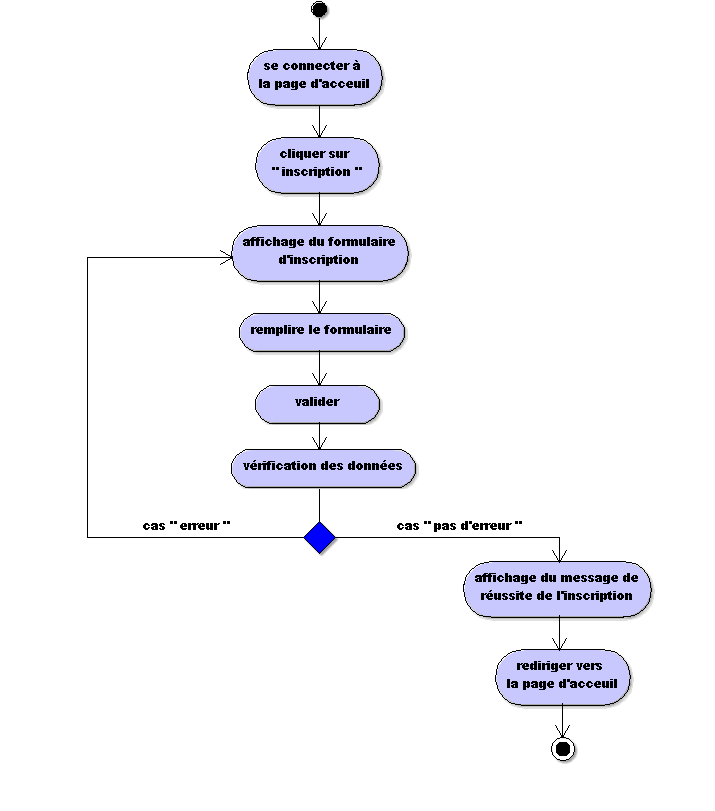
****

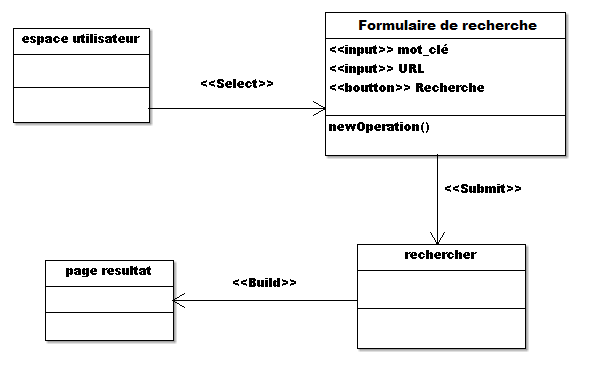
Figure II.9 : Diagramme d’activité« inscription »

**4.3. Les diagrammes de classe :**

Représentent la vue statique des objets pages, leurs intérêt majeur est de modéliser les entités d’un système. Autrement dit, ils expriment les relations existantes entre les pages client et serveur.

**4.3.1. Les diagrammes de classe détaillés :**

**4.3.1.1. Diagramme de classe détaillé du cas d’utilisation «rechercher une chronique » :**

****

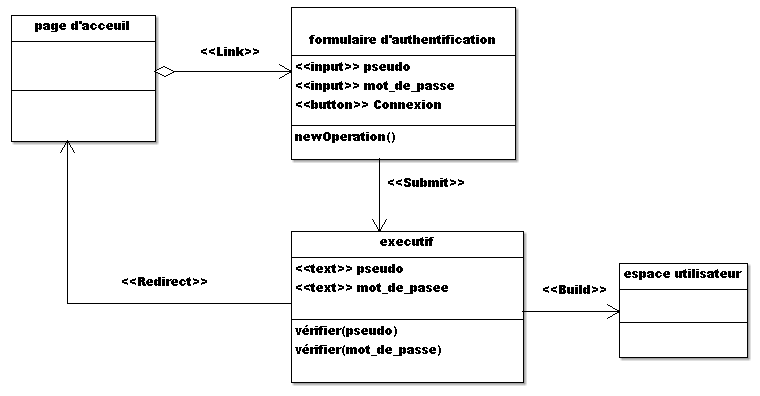
**Table**

**chronique**

Figure II.10 : Diagramme de classe détaillé

«Rechercher une chronique »

**4.3.1.2. Diagramme de classe détaillé du cas d’utilisation «Authentification» :**

****

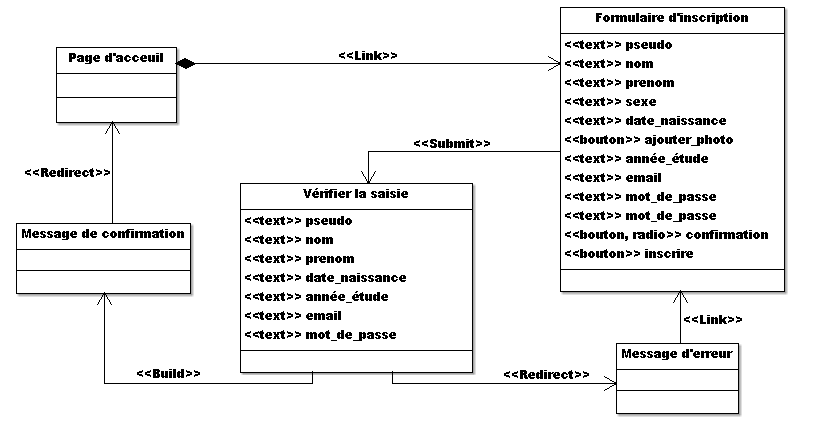
**Table**

**utilisateur**

Figure II.11 : Diagramme de classe détaillé « Authentification »

**4.3.1.3. Diagramme de classe détaillé du cas d’utilisation «inscription » :**

**executif**

****

**Table**

**utilisateur**

Figure II.12 : Diagramme de classe détaillé «Inscription »

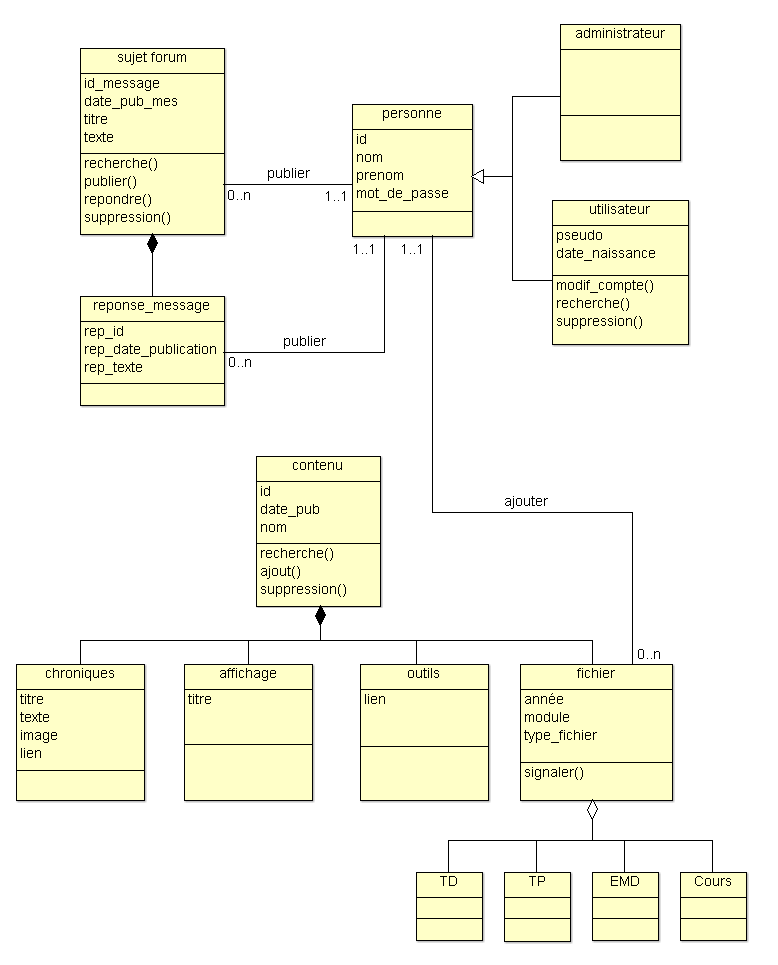
**4.3.2. Le diagramme de classe global :** ****

Figure II.13 : Diagramme de classe global

**4.4. Modèle entité-association :**

Le modèle entité-association est l’un des modèles conceptuels les plus utilisés pour construire une base de données. Il fournit un formalisme graphique appelé diagramme entité-association très simple qui facilite la modélisation de données.

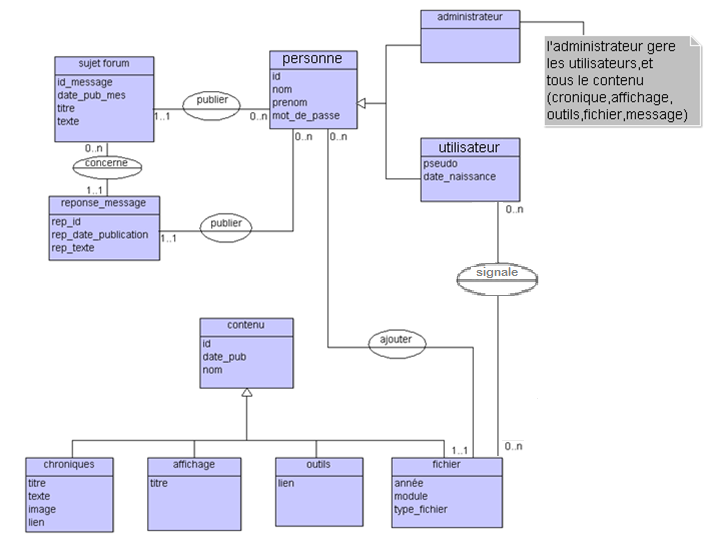
Le modèle entité-association est généralement fondé sur trois principaux concepts (l’entité, l’association et la propriété) qui permettent de décrire un ensemble de données relatives à un domaine défini.

Figure II.14 : Diagramme entité-association

**5. Conclusion :**

Au début de ce chapitre, nous avons suivi le langage de modélisation UML pour l’étude du futur système (analyse et conception), pour spécifier les cas d’utilisations, concevoir les diagrammes de séquence, d’activité et élaborer les digrammes de classes.

A la fin de ce chapitre, nous avons défini le modèle entité-association accompagné avec le diagramme spécifié. L’implémentation étant la partie qui suit celle de la conception qui nous permet la réalisation de notre application, les différents outils de l’environnement de développement et de fonctionnement de notre système seront l’objet du chapitre prochain.