**Chapitre II : Analyse Et Conception**

# Introduction :

La conception est une étape primordiale dans le cycle de vie d’une application, elle a pour objectif d’élaborer à partir du modèle du système obtenu lors de l’étape d’analyse de besoin, des modèles détaillés de l’architecture du système. Elle vise également la réduction de la complexité du système.

Ce chapitre sera consacré à la conception de notre projet en utilisant le langage de modélisation UML. Mais avant tout, nous allons exposer quelques notations d’UML tels que les diagrammes et les modèles.

# Développement orienté objet :

Les caractéristiques de l’objet répondent invariablement par les avantages apportés qui sont stabilité de la modélisation par rapport aux entités du monde réel, la construction itérative facilitée par le couplage faible entre composants et la possibilité de réutiliser des éléments de développement d’un autre objet. Certains insistent également sur la simplicité du modèle qui ne fiat appel qu’à cinq concepts fondamentaux (les objets, les messages, les class, l’héritage et le polymorphisme) pour exprimer de manière uniforme l’analyse, la conception et la réalisation d’une application informatique. Tous ces points sont parfaitement exacts, mais ils ne sont que la conséquence de la capacité fantastique d’intégration de l’approche objet [1].

# Présentation générale d’UML :

## Historique :

Le langage de Modélisation Unifié (UML) est le résultat de la combinaison des trois principales méthodes orientées objet : BOOCH, OMT et OOSE.

OMT (*Object Modeling Technique*) de James Rumbaugh et BOOCH de Grady Booch ont été les deux méthodes les plus diffusées durant les années 90.Par ailleurs, OOSE de Ivar Jacobson s’est aussi imposée dans le monde objet pour la partie formalisation des besoins. Ces trois auteurs ont ensuite décidé d'unir leurs efforts au sein de la société Rational Software et en 1996 la première version d'UML est proposée [2].

Les grandes étapes de la diffusion d’UML peuvent se résumer comme suit [3] :

* 1994-1996 : rapprochement des méthodes OMT, BOOCH et OOSE et naissance de la première version d’UML.
* 23 NOVEMBRE 1997 : version 1.1 d’UML adoptée par l’OMG.
* 1998-1999 : sortie de versions 1.2 à 1.3 d’UML.
* 2000-2001 : sortie des dernières versions suivantes 1.x.
* 2002-203 : préparation de la V2.
* 10 OCTOBRE 2004 : sortie de la version 2.1.
* MAI 2010 : sortie de la version 2.3.
* JUIER 2011 : sortie de la version 2.4.1.
* DECEMBRE 2017 : sortie de la version 2.5.1.

## Définition :

UML, c’est l’acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un langage visuel constitué d’un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d’être effectuées par le logiciel, etc. [4].

## Les diagrammes UML :

UML dans sa 2ème version propose treize diagrammes qui peuvent être utilisés pour la description d’un système. Ces diagrammes sont regroupés dans deux grands ensembles qui sont :

### **Les diagrammes structurels** [5] :

Qui  ont comme  vocation de représenter l’aspect statique d’un système. Ils permettent d’identifier les objets constituant le programme, leurs attributs, leurs opérations et les méthodes qui leurs sont associés. Ils sont au nombre de six à savoir :

* Diagramme de Class : Le diagramme de classe constitue l’un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d’association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n’est pas représenté directement dans le diagramme de classe ; seul l’algorithme général et le pseudocode correspondant peuvent être associés à la modélisation.

La description de diagramme de class est fondée sur les concepts d’objet, de class et les différents type d’association entre classes.

* Diagramme d’objet : il sert à représenter les instances de classes (objets) utilisées dans le système.
* Diagramme de composant : il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...).
* Digramme de déploiement : il sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent avec eux.
* Diagramme de paquetage : permet de représenter la hiérarchie des paquetages du projet, leur organisation et leurs interdépendances, simplifie les diagrammes (donc plus simple à comprendre).
* Diagramme de structure composite : permet de décrire la structure interne d'un objet complexe lors de son exécution (c'est à dire, décrire l'exécution du programme), dont ses points d'interaction avec le reste du système.

### Les diagrammes de comportement [6] :

Ces diagrammes représentent la partie dynamique d’un système régissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept Diagrammes sont proposés par UML 2 :

* Diagramme de cas d’utilisation : il décrit les possibilités d'interaction entre le système et les acteurs, c'est-à-dire toutes les fonctionnalités que doit fournir le système.
* Diagramme d’états-transition : il montre la manière dont l'état du système (ou de sous-parties) est modifié en fonction des événements du système.
* Diagramme d’activités : variante du diagramme d'états-transitions, il permet de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (multithreads ou multiprocessus).
* Diagramme de séquence : la représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou des acteurs.
* Diagramme de communication : la représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
* Diagramme global d’interaction : variante du diagramme d'activité où les nœuds sont des interactions, permet d'associer les notations du diagramme de séquence à celle du diagramme d'activité, ce qui permet de décrire une méthode complexe.
* Diagramme de temps : la représentation des interactions où l'aspect temporel est mis en valeur, il permet de modéliser les contraintes d'interaction entre plusieurs objets, comme le changement d'état en réponse à un évènement extérieur.

# Présentation de notre projet :

Notre projet consiste à concevoir un site web pour le suivi et l’évaluation des projets de fin d’étude (PFE). Ce site est pour but de facilité l’échange des documentations entre les différents utilisateurs de site. Notre site est découpé en deux parties :

1. Espace Membre : cet espace est dédié au utilisateur au ils peuvent :

* Consulter les documents PFE.
* Rechercher un document PFE.

Il y’a la possibilité d’inscrire sur le site pour accéder au d’autre fonctionnalité comme :

* Evaluer un document PFE.
* Mettre un commentaire sur un document PFE.
* Ajouter un document PFE.
* Gestion de profil.
* Télécharger un document PFE.

1. Espace Administration : c’est l’espace au l’administrateur de site peut faire le suivant :

* Gestion d'utilisateurs
* Gestion des documents
* Gestion des profils.
* Validation des demandes d’inscription.
* Gestion des archives.

Notre site web sera sécurisé à l’aide d’un système d'authentification.

# Modélisation de notre projet :

Notre conception va accumuler et étudier quatre diagrammes qui sont : diagrammes de cas d’utilisation, classe, séquence et d’activités.

## Les diagrammes de cas d’utilisation :

L’objectif fondamental de cette étape est d’identifier les principaux cas d’utilisation. Nous nous intéressons donc, dans cette partie, à la réalisation des diagrammes des cas d’utilisations. Ces diagrammes décrivent précisément les besoins du client final et spécifient le comportement attendu par le système à développer. Généralement un diagramme de cas d’utilisation modélise un service rendu par le système.

### Présentation des acteurs :

Avant d’étudier les cas d’utilisation nous commençons par la recherche des acteurs qui régissent notre champ d’étude.

Dans notre site web, nous avons défini trois acteurs majeurs qui sont :

* Visiteur de site (est une personne qui n’admet pas d’un compte sur le site)
* Membre de site (étudiant, enseignant ayant un compte sur le site).
* Administrateur

### Identification des cas d’utilisation :

Les différents cas d’utilisations sont expliqués dans le tableau suivant :

|  |  |
| --- | --- |
| **Libellé** | **Description** |
| **Création d’un compte** | Un visiteur peut inscrire au site |
| **Rechercher un document** | Rechercher les documents par mot-clé ou par spécialité |
| **Consulter un document** | Affichage de document avec leur information associée |
| **Ajouter un document** | Charger un document sur le serveur |
| **Gestion de profil** | Changer les informations personnelles d’un membre |
| **Gestion des commentaires** | Ajouter ou supprimer des commentaires |
| **Gestion des documents** | Ajouter ou supprimer ou modifier un document |
| **Gestion d’utilisateurs** | Ajouter ou supprimer un membre |

Tableau 1 : Liste des cas d'utilisation

### Diagramme de cas d’utilisation d’un visiteur :

Le visiteur peut créer un compte, et pour finaliser l’inscription, l’administrateur valide la demande qu’il est un vrai étudiant ou enseignant. Le visiteur peut aussi consulter et rechercher un document sons avoir un compte dans le site.

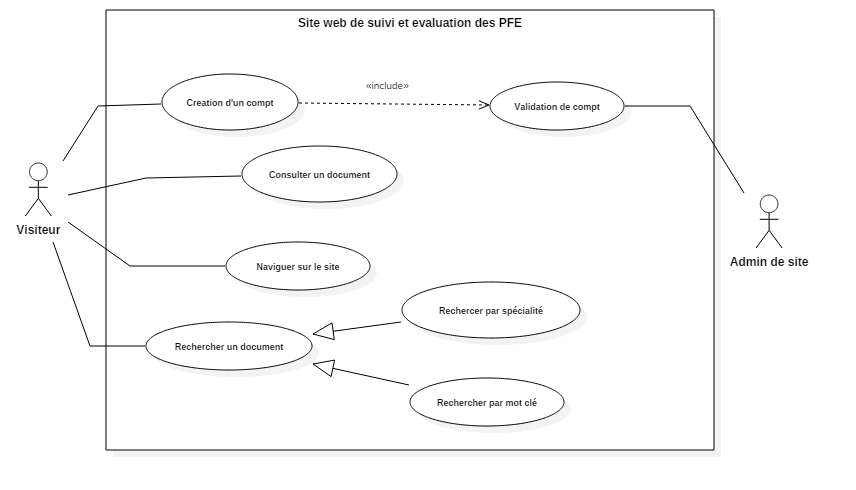
****

Figure 1 : Diagramme de cas d'utilisation d'un visiteur

### Diagramme de cas d’utilisation d’un membre :

Un membre a la possibilité au d’autre fonctionnalité de site, comme il peut ajouter des évaluations ou des commets sur les documents et les télécharger, ainsi, il peut gérer leur compte et profil et faire rechercher un document soit par mot-clé ou par spécialité.

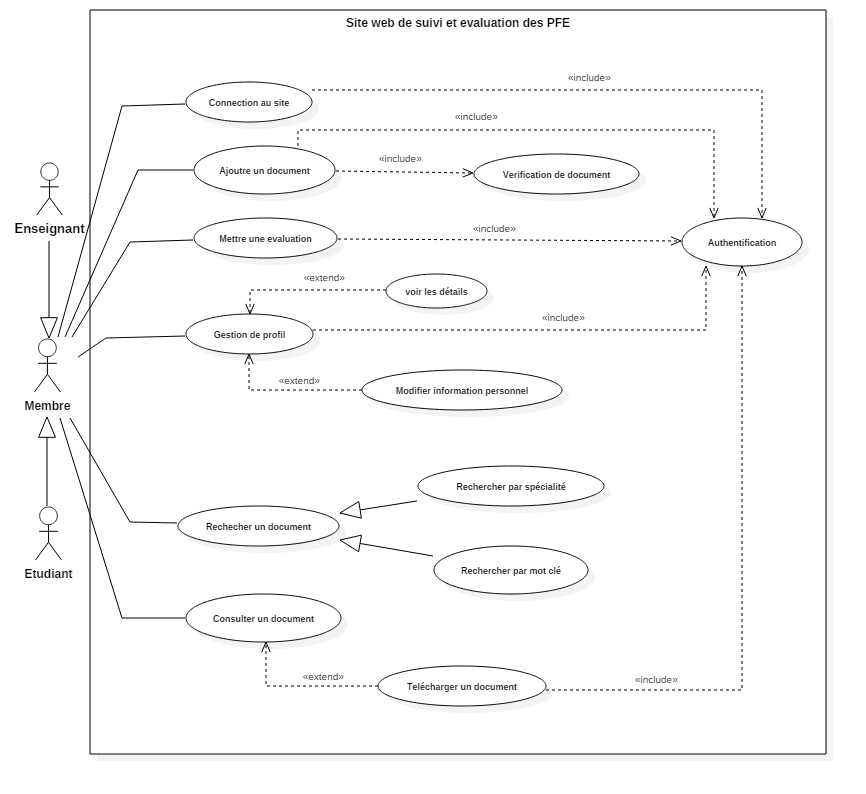
****

Figure 2: Digramme de cas d'utilisation d'un membre

### Diagramme de cas d’utilisation d’un administrateur :

L’administrateur de site est le responsable de la gestion générale de site, il fait la gestion d’utilisateur (ajoute, suppression et modification) et la gestion des documents et la validation des comptes (cas de nouvelle inscription), aussi, il peut consulter l’archive.

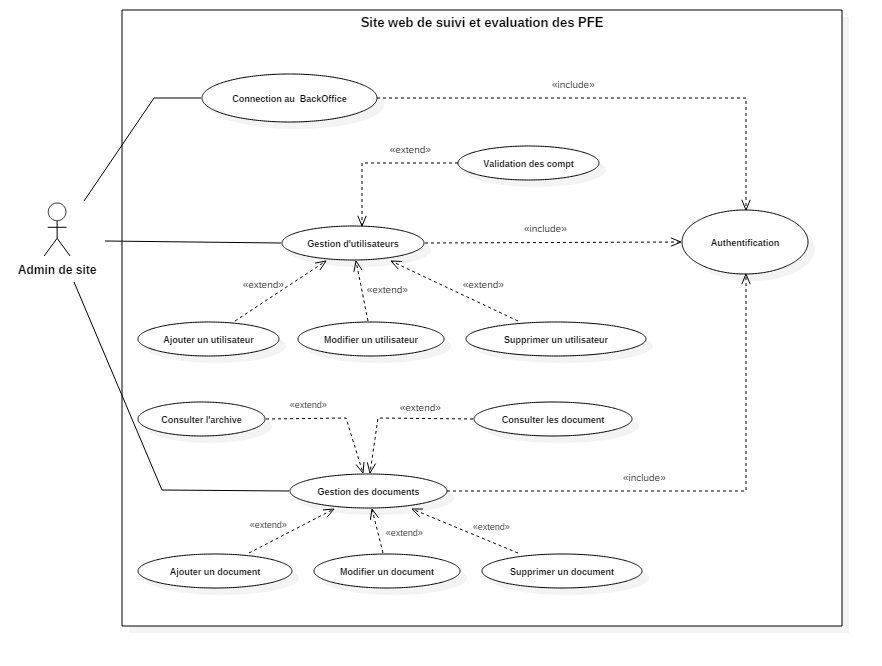
****

Figure 3: Diagramme de cas d'utilisation d'un admin

## Le diagramme de class :

Une classe représente la structure d'un objet, c'est-à-dire la déclaration de l'ensemble des entités qui le composent. Elle est constituée d'attributs dont les valeurs représentent l'état de l'objet et des méthodes qui sont les opérations applicables aux objets.

### Description de notre diagramme de class :

Notre digramme comporte de huit class, sont expliquées dans le tableau suivante:

|  |  |
| --- | --- |
| Class | Description |
| Visiteur | Une classe qui représente un visiteur de site avec les fonctions qu’il peut faire, comme :   * Créer un compte * Rechercher un document * Consulter un document |
| Membre | Une classe qui représente un membre de site avec leurs attributs et opérations |
| Admin | Une classe qui représente un administrateur de site avec leurs attributs et opération |
| Document | Une classe qui définit les attributs d’un document électronique |
| Evaluation | Une classe qui définit les attributs d’un d’une évaluation |

Tableau 2: Description des class

Notre site permet au membre d’ajouter ou consulter des documents de type PFE, et chaque document a la possibilité d’avoir des évaluations. Un membre peut être un étudiant ou un enseignant.

L’administrateur vise à tenter de suivre le site. Il contrôle le site par la gestion des documents et des membres et leurs évaluations.

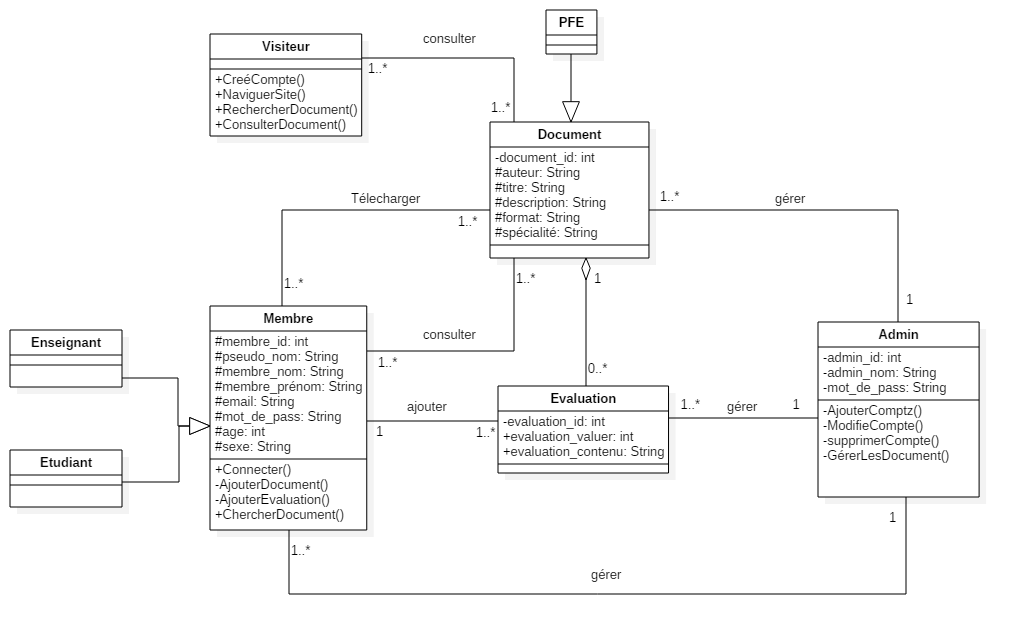
La figure ci-dessous présente le diagramme de classes relatif au projet : 

Figure 4 : Digramme de class de projet

## Les diagrammes de séquence :

Les diagrammes de séquences sont 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. Dans un souci de simplification, on représente l’acteur principal à gauche du diagramme et les acteurs secondaires éventuels à droite du système.

Le but étant de décrire comment se déroulent les actions entre les acteurs ou les objets. Les périodes d’activité des classes sont symbolisées par des rectangles.

### Diagramme de séquence d’inscription :

Dans la figure suivante, on a un diagramme de séquence qui décrit séquentiellement la dynamique entre les objets : Visiteur, Site web et la base des données pour exécuter la tâche d’inscription. Cette dynamique est basée sur :

* Un visiteur remplie leur information.
* Le site valide leurs données et les sauvegarde dans la base de donné.

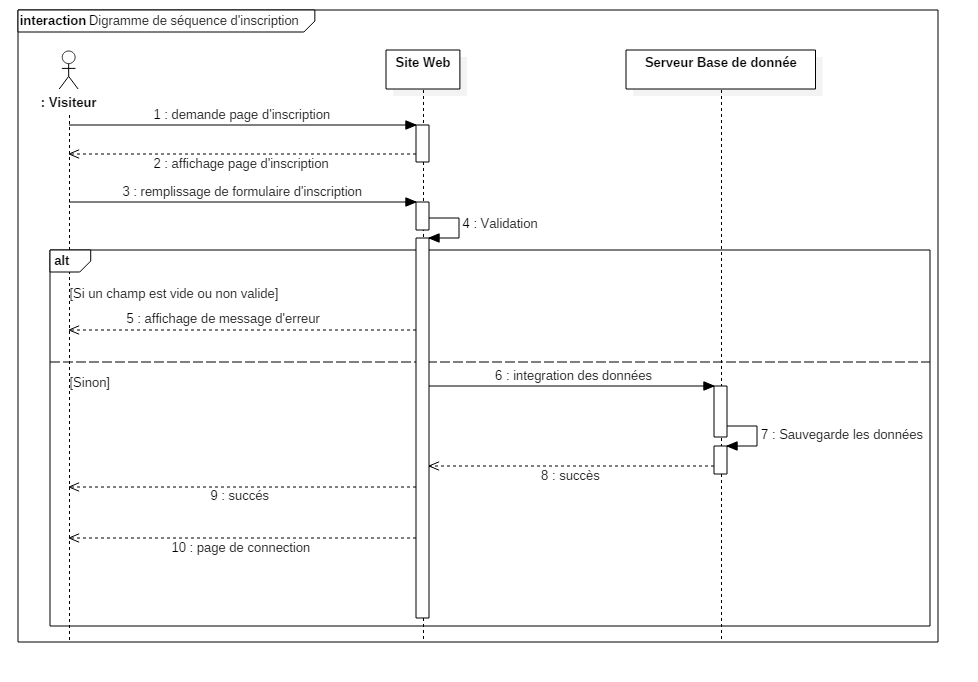


Figure 5 : Diagramme de sequence d'inscription

### Diagramme de séquence d’authentification :

Le diagramme ci-dessous définit la dynamique de phase d’authentification. D’abord, le membre demande l’accès au site par l’email et le mot de passe. Après la vérification du compte, le site répond par afficher la page d’accueil ou un message d’erreur.

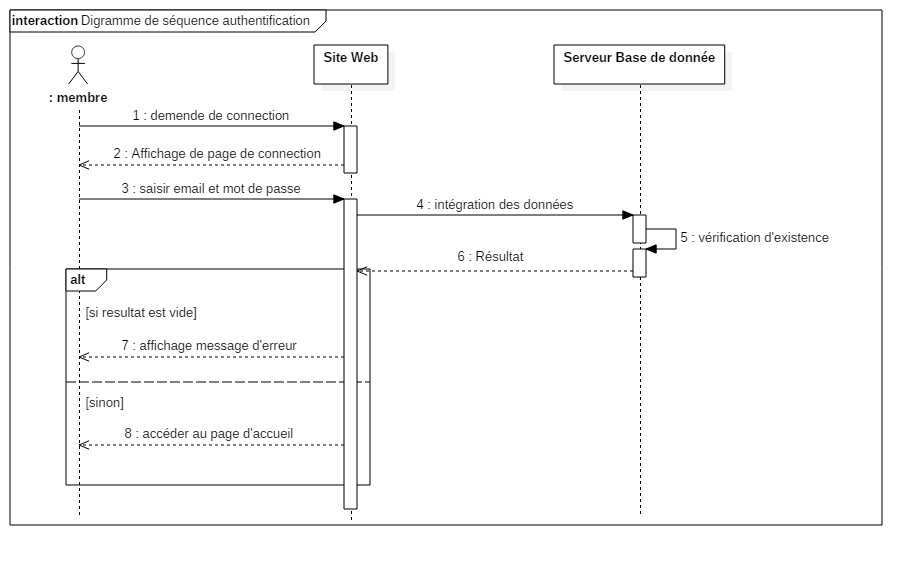
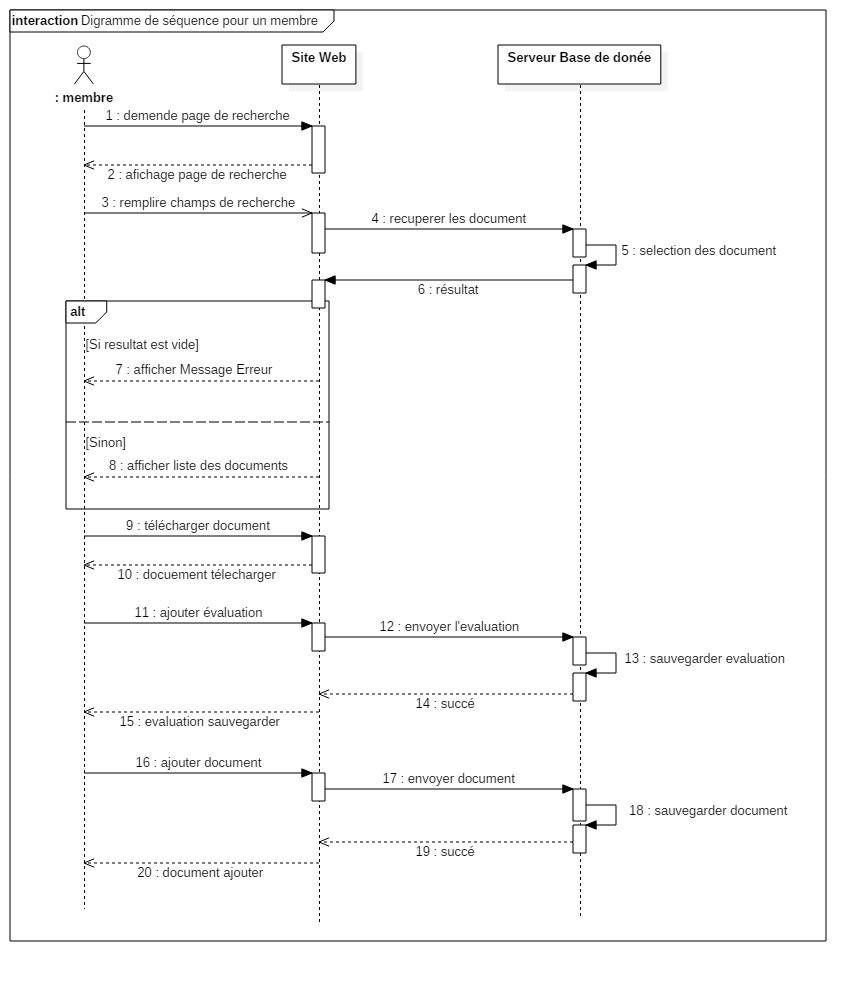


Figure 6 : Diagramme de séquence d'authentification

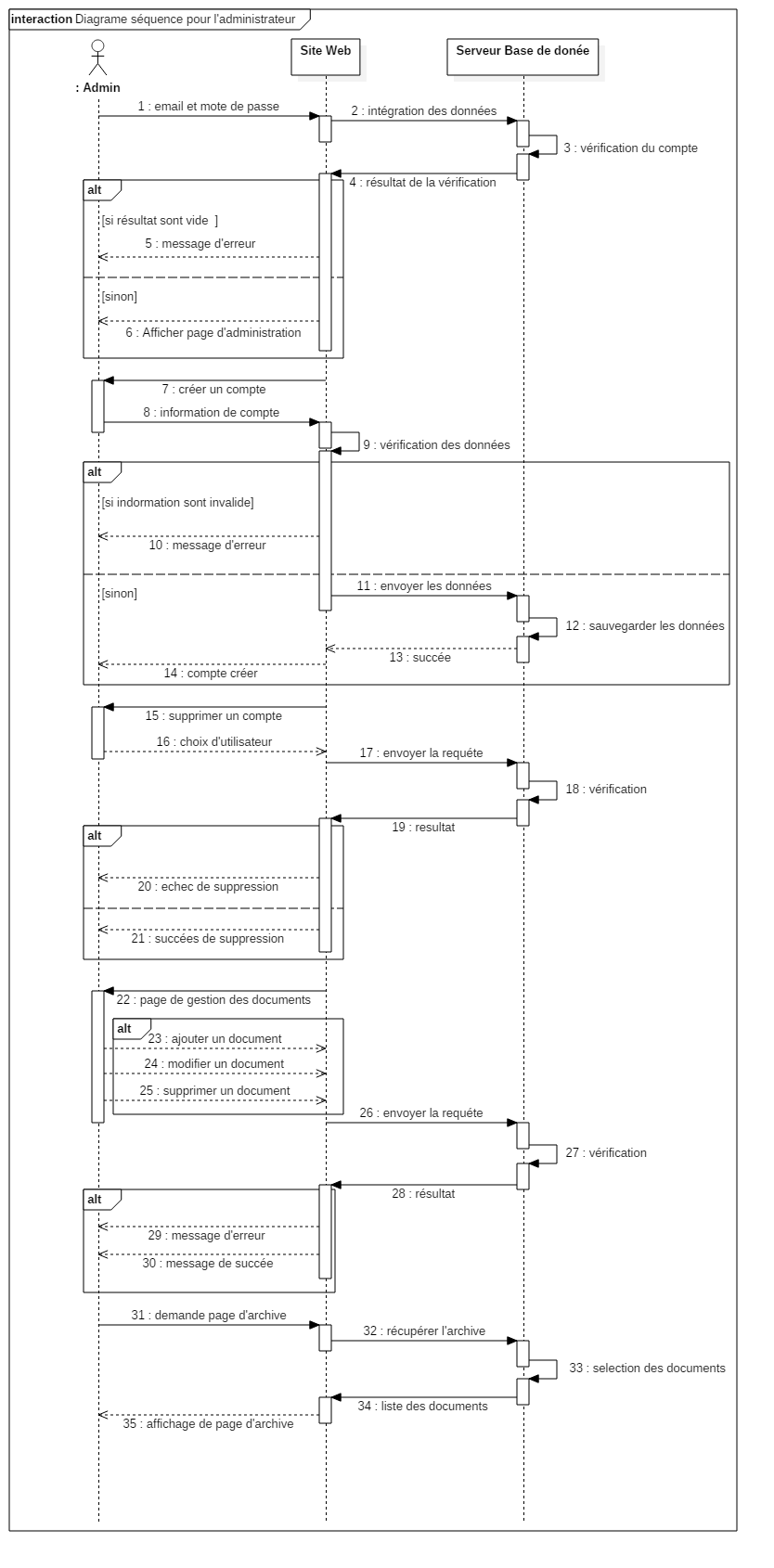
### Diagramme de séquence pour un membre :

Description à faire…



### Diagramme de séquence pour l’administrateur :

Description à faire…



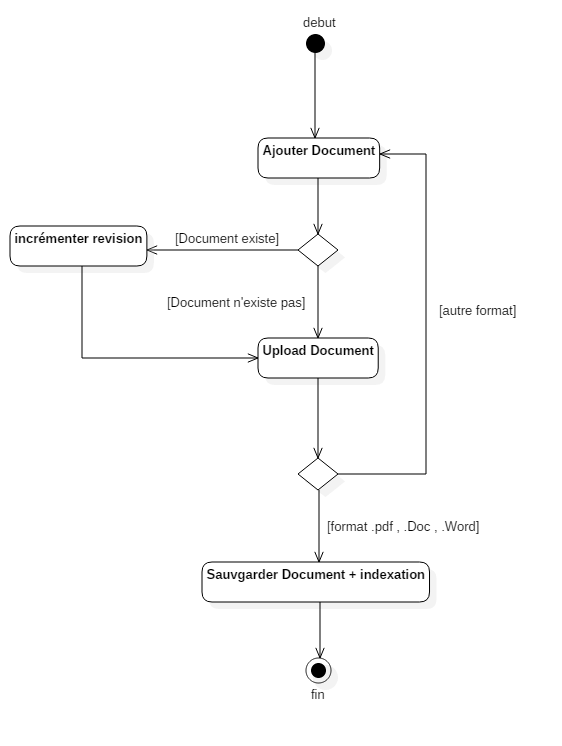
## Les diagrammes d’activité :

Le diagramme d’activités permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d’utilisation.

Les deux figures suivantes illustres les diagrammes d’activités de l’ajout et le téléchargement des documents.

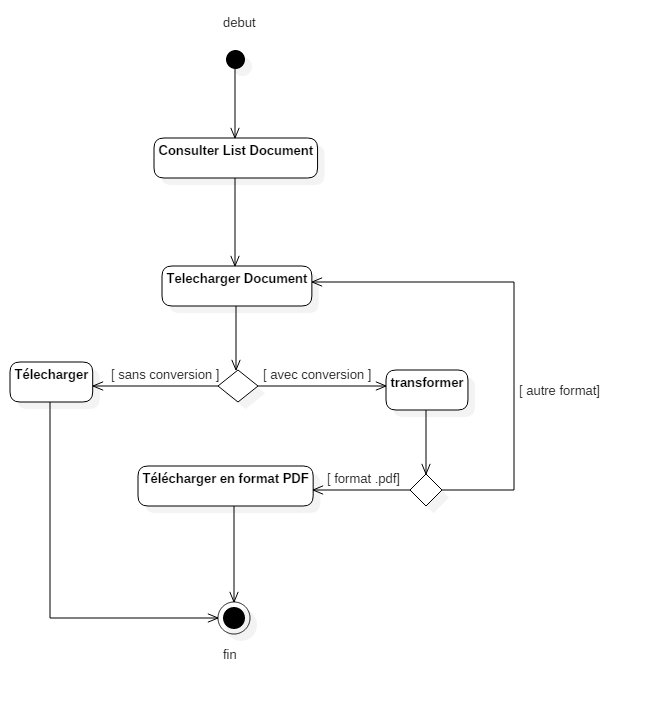
### Diagramme d’activité «  Ajouter document  » :

Lors de l’ajout d’un document, on commence par vérifier s’il est déjà présent dans le serveur. Dans ce cas, la révision est incrémentée. En accepte seulement les documents de format .pdf, .word, .doc. Enfin, on sauvegarde le document et on l’index.



### Diagramme d’activité «  Télécharger document  » :

Lors du téléchargement d’un document, l’utilisateur peut choisir plusieurs formats. Au cas où il choisit un format autre que l’original, le document sera converti.



# Conclusion :

Dans ce chapitre, nous focalisons sur la partie conception de notre Project. On a essayé d’expliquer les différents concepts de langage de modélisation UML, ainsi, que les différents diagrammes de notre site web.

Le chapitre suivant concerne la partie réalisation de notre site web.

# Références : « les référence ne sont pas en ordre (chapitre 1 + 2), il faut combiner tous les chapitre dans un document »

|  |  |
| --- | --- |
| [1] | M. PIERRE-ALAIN et G. NATHALIE, Modélisation object avec UML, Eyrolles, 2004. |
| [2] | G. JOSEPH et G. DAVID, UML 2 Analyse Et Conception, Paris: Dunod, 2008. |
| [3] | «omg.org,» OBJECT MANAGEMENT GROUP, [En ligne]. Available: https://www.omg.org/spec/UML/About-UML/. [Accès le 08 02 2019]. |
| [4] | «openclassrooms.com,» OpenClassRooms, [En ligne]. Available: https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2035851-uml-c-est-quoi. [Accès le 08 02 2019]. |
| [5] | S. OLIVIER, Introduction à la modélisation orientée objets avec UML, 2010. |
| [6] | B. XAVIER et M. ISABELLE, UML 2 pour les développeyrs, Eyrolles, 2006. |
| [7] | K. Sameh, «GESTIONET ARCHIVAGE DE DOCUMENTS ELECTRONIQUES : EVIDENCE, FIABILITE ET AUTHENTICITE,» 1998. |
| [8] | L. Riffi, L'Archivage Electronique : ce qu'il faut savoir, Micro-Plus. |
| [9] | J. Lassoury, La documentique : getion élétronique de documents et gestion documentairs, Paris: Dunod. |
| [10] | D. Géraldine, L. Lucile et L. Anne-Gaelle, «La Gestion Electronique des documents,» 2008. |
| [11] | N. Benslimane, « La Gestion Électronique des Documents : un système au secours de l'information,» Micro-Plus. |
| [12] | « Bien comprendre les fonctionnalités d'une GED,» StarXpert, LYON , 2011. |
| [13] | M. Hudon, Introduction à l'indexation, à la classification et à la condensation des documents, Press de l'Université du Québec, 2013. |
| [14] | J. CHAUMIER et S. CACALY, Dictionnaire encyclopédique de l’information et de la documentation, Nouvelle édition éd., Paris, 2001. |
| [15] | «appvizer.fr,» [En ligne]. Available: https://www.appvizer.fr/magazine/collaboration/gestion-documentaire-ged. [Accès le 12 02 2019]. |
| [16] | CENTRE INFORMATIQUE NATIONAL DE L'ENSEIGNEMENT SUPERIEUR, [En ligne]. Available: https://www.cines.fr/archivage/un-concept-des-problematiques/le-modele-de-reference-loais/. [Accès le 12 02 2019]. |
| [17] | OPEN SOURCE GUIDE by Smile, [En ligne]. Available: http://www.open-source-guide.com/Solutions/Applications/Ged-ecm. [Accès le 12 02 2019]. |
| [18] | C. I. D. ARCHIVES, «GUIDE POUR LA GESTION ARCHIVISTIQUE DES DOCUMENTS ELECTRONIQUES,» FÉVRIER 1997. [En ligne]. Available: https://www.ica.org/sites/default/files/ICA%20etude%208\_Fr.pdf. [Accès le 10 02 2019]. |
| [19] | M. Auffret, «L’archivage pérenne des,» CINES. |
| [20] | «Recommandations relatives à la conception et à l'exploitation de systémes informatiques en vue d'assurer la conservation er l'intégrité des documents stockés dans ces systémes,» ANFOR, 2001. |
| [21] | M. PIERRE-ALAIN et G. NATHALIE, Modélisation object avec UML, Eyrolles, 2004. |