

**Sujet :**

# **LA MISE EN ŒUVRE D'UNE APPLICATION DE SUIVI DES STAGES**

Elaboré par :

**Mohamed HAMDI  
&  
Mohamed REZGUI**

## **RAPPORT**

**De**

**Projet de fin d'Etudes**

Présenté en vue de l'obtention du diplôme  
De

**LICENCE APPLIQUEE EN SCIENCES ET TECHNIQUES DE  
L'INFORMATION ET DE COMMUNICATIONS**

**Encadré par :**

Mr Riadh BOUHOCHI

**Société d'accueil : Université Virtuelle de Tunis**

**Année Universitaire : 2010/2011**

# *Dédicaces*

*Nous dédions ce projet de fin d'études en témoignage de  
notre gratitude et de tous nos respect à :*

*Nos parents pour toute l'affection, tous les sacrifices et  
les encouragements qu'ils n'ont pas cessé de nous  
prodiguer tout au long de nos études.*

*A tous ceux qui nous ont soutenus tout au long de ce  
projet.*

*A notre encadreur Riadh Bouhouch et  
A nos enseignants,*

*Mohamed Hamdi  
&  
Mohamed Rezgui*

## **Remerciement**

*Nous voulons exprimer par ces quelques lignes de remerciements notre gratitude envers tous ceux en qui, par leur présence, leur soutien, leur disponibilité et leurs conseils nous avons trouvé courage afin d'accomplir ce projet.*

*Nous commençons par remercier Monsieur Riadh BOUHOUCHE qui nous a fait l'honneur d'être notre encadreur. Nous le remercions profondément pour son encouragement continu et aussi d'être toujours là pour nous écouter, nous aider et nous guider à retrouver le bon chemin par son sagesse et ses précieux conseils, ce qui nous a donné la force et le courage d'accomplir ce projet.*

*Nous tenons à remercier également Monsieur Belgacem Hrizi pour son aide et soutien.*

*En fin, nous ne pouvons achever ce projet sans exprimer notre gratitude à tous les enseignants de l'Université Virtuelle de Tunis, pour leur dévouement et leur assistance tout au long de notre formation.*

## تلخيص

مشروعنا يتلخص في دراسة و تصور و تحقيق منظومة واب لمراقبة التربصات لمصلحة الجامعة الافتراضية بتونس و هي تتيح تطوير الاتصال و تبادل المعلومات و الملفات بين الطالب و المؤسسة التربوية و المؤسسة المضيفة و تتيح أيضا تقييم و مراقبة التربصات.

تتمثل أهدافنا الرئيسية من هذا المشروع في إدارة متابعة ومراقبة التربصات. لتحقيق أهدافنا قمنا بإنشاء منظومة واب تعتمد تقنية ثلاثي الطبقات مع استخدام "ماي سكل" كنظام إدارة قواعد البيانات و استخدام لغة النمذجة "إم ل" و "ماي إكليريس" كمحيط تطوير. وأخيرا، المنظومة المحققة سوف يتم إستغلالها من قبل طلبة و مدرسي الجامعة الافتراضية بتونس و المؤسسات المضيفة للتربصات. إن هذا التقرير يصف مراحل تحقيق هذا المشروع. الكلمات المفتاحية: "أسس", "ماي سكل", "إم ل", "أه م", "ري ب", "ج د ك".

## Résumé

Notre projet consiste dans l'étude, la conception et la réalisation d'une application de suivi des stages pour le compte de l'Université Virtuelle de Tunis, qui permet l'amélioration de la communication et l'échange de l'information et de la documentation entre l'environnement d'accueil et l'université, ainsi que l'évaluation et le contrôle numérique des stages.

Les objectifs majeurs de cette application sont la gestion, le suivi et le contrôle des stages. Pour atteindre cet objectif, nous avons eu recours à créer une plateforme web basé sur une architecture 3 tiers dont MySQL comme système de gestion de bases de données, une étude conceptuel réalisé par le langage de modélisation UML et MyEclipse comme IDE.

Enfin, notre application sera exploitée par les étudiants et les enseignants de l'Université Virtuelle de Tunis.

Le présent rapport décrit les différentes étapes de réalisation de ce projet.

**Mots-clés :** ASS, MySQL, UML, IHM, RUP, JDK, JEE, IDE, API, HTML, CSS, JSP.

## Abstract

Our project consists in the conception and the realization of a management Application of the follow-up of the traineeship, which allows the communication and the exchange of information and documentation between students and preceptors and evaluation and control of the traineeship.

The major objectives of this application are the management, follow-up and control of the traineeship.

To reach this objective, we had appeal to create a web platform with MySQL as database management system and UML as modeling language.

Finally, our application will be exploited by students and teachers of the Virtual University of Tunis.

The present report describes the various stages of realization of this project.

**Keywords:** ASS, MySQL, UML, IHM, Modèle en V, RUP, JDK, JEE, IDE, API, HTML, CSS, JSP.

# Table des matières

## **INTRODUCTION GÉNÉRALE----- 1**

## **CHAPITRE 1 : SPÉCIFICATION ET ANALYSE DES BESOINS ----- 2**

### **INTRODUCTION ----- 2**

#### **I. ETUDE DE L’EXISTANT ----- 2**

1. Description de l’existant ----- 2

2. Critique de l’existant ----- 2

3. Solution proposée----- 3

#### **II. ETUDE DES BESOINS ----- 3**

1. Besoins fonctionnels ----- 3

a. L’échange de l’information et de la documentation ----- 4

b. La gestion du cahier du stage ----- 4

c. Le Forum ----- 4

d. La gestion du BackOffice ----- 4

2. Besoins non fonctionnels ----- 4

#### **III. CADRE DE PROJET ----- 5**

### **CONCLUSION ----- 6**

## **CHAPITRE 2 : CONCEPTION ----- 6**

### **INTRODUCTION ----- 7**

#### **I. CONCEPTION GÉNÉRALE ----- 7**

1. Le modèle de cycle de vie ----- 7

a. Présentation du model----- 7

b. Description du modèle ----- 8

2. Méthodologie adoptée----- 8

3. Concept et architecture de l’AGSS (Application de Gestion et Suivi des Stages) ----- 9

#### **II. CONCEPTION DÉTAILLÉ----- 10**

1. Les diagrammes des cas d’utilisation ----- 10

a. Présentation des acteurs-----	10
b. Description des cas d'utilisation-----	11
2. Les diagrammes d'activité-----	20
a. Diagramme d'activité Affectation des tâches par l'encadreur -----	20
b. Diagramme d'activité Remplir le cahier de stage -----	21
c. Diagramme d'activité Ajouter un stage-----	22
3. Les diagrammes de séquences -----	23
a. Diagramme de séquence d'authentification -----	23
b. Diagramme de séquence déposer un document -----	24
c. Diagramme de séquence valider un stage -----	25
d. Diagramme de séquence ajouter un stage -----	26
4. Diagramme des classes -----	27
a. Description des classes-----	27
b. Le diagramme des classes -----	29
c. Modèle relationnel -----	30
<b>III. MAQUETTE -----</b>	<b>31</b>
1. Structure de l'application -----	31
2. Charte graphique-----	31
<b>CONCLUSION -----</b>	<b>32</b>

## **CHAPITRE 3 : RÉALISATION-----33**

<b>INTRODUCTION -----</b>	<b>33</b>
<b>I. ENVIRONNEMENT DU TRAVAIL -----</b>	<b>33</b>
1. Environnement matériel -----	33
2. Atelier de génie logiciel-----	33
a. Langage de programmation -----	33
b. Environnement de développement -----	33
c. Conception -----	34
d. IDE (Environnement de développement intégré) -----	35
e. Serveur d'Application-----	35
f. Système de gestion de base des données-----	35
g. Framework -----	36
h. Design et MultiMedia-----	37

<b>II. SCÉNARIOS APPLICATIF</b>	<b>38</b>
1. Espace Administrateur « BackOffice »	39
a. Page d'accueil	39
b. Interface « Ajouter un stage »	39
2. Espace Etudiant	41
a. Page d'accueil	41
b. Interface « Déposer un document »	42
3. Espace Encadreur	43
a. Page d'accueil	43
b. Interface « Affecter une tâche »	43
4. Espace Membre du Jury	44
a. Page d'accueil	44
b. Interface « Valider un stage »	44
<b>CONCLUSION</b>	<b>45</b>
<b>CONCLUSION ET PERSPECTIVES</b>	<b>46</b>

## Liste des figures

Figure 1: Schéma présentatif du cadre de projet -----	5
Figure 2: Le model en V -----	7
Figure 3: La structure générique d'une architecture 3-tiers -----	9
Figure 4: Diagramme de cas d'utilisation de l'étudiant -----	12
Figure 5: Diagramme de cas d'utilisation de l'encadreur -----	15
Figure 6: Diagramme de cas d'utilisation du membre de jury -----	17
Figure 7: Diagramme de cas d'utilisation de l'Administrateur -----	18
Figure 8: Diagramme d'activité Affectation des tâches par l'encadreur -----	20
Figure 9: Diagramme d'activité Remplir le cahier de stage -----	21
Figure 10: Diagramme d'activité Ajouter un stage -----	22
Figure 11: Diagramme de séquence d'authentification -----	23
Figure 12: Diagramme de séquence déposer un document -----	24
Figure 13: Diagramme de séquence valider un stage -----	25
Figure 14: Diagramme de séquence ajouter un stage -----	26
Figure 15: Diagramme des classes -----	29
Figure 16: Structure du site en évolution -----	31
Figure 17: Charte graphique -----	31
Figure 18: Authentification -----	38
Figure 19 : Page d'accueil BackOffice -----	39
Figure 20 : Interface de l'Ajout de stage -----	39
Figure 21: Interface d'affectation des étudiants -----	40
Figure 22: Interface d'affectation des encadreur -----	40
Figure 23: Fiche Stage -----	41
Figure 24: Page d'accueil de l'étudiant -----	41
Figure 25: Interface de Dépôt d'un document -----	42
Figure 26: Interface de MAJ d'un document -----	42
Figure 27: Page d'accueil de l'encadreur -----	43
Figure 28: Interface d'affectation des tâches -----	43
Figure 29: Page d'accueil du membre de jury -----	44
Figure 30: Interface de sélection du stage -----	45
Figure 31: Interface de validation du stage -----	45



## **Introduction générale**

Les espaces numériques de travail (ENT) sont aujourd'hui largement préconisés dans les établissements universitaires. Basés sur l'utilisation des technologies de l'information et de la communication, ils désignent un dispositif global fournissant à un usager un point d'accès à travers les réseaux à l'ensemble des ressources, des applications, des outils et des services numériques à partir d'un portail unique en rapport avec son activité. Il est un point d'entrée pour accéder au système d'information de l'établissement.

Dans ce contexte, nous avons proposé de lancer un projet qui consiste à développer une application de suivi des stages qu'elle va être considérée comme un espace numérique de travail pour le compte de l'Université Virtuelle de Tunis.

L'objectif de ce projet est d'analyser, concevoir et développer un espace numérique de suivi des stages. Cet espace doit permettre, en premier lieu, l'échange de l'information et de la communication entre les étudiants et les encadreurs, le contrôle, le suivi et l'évaluation des stages. En deuxième lieu, l'application doit être dotée d'une partie BackOffice qui permettra de paramétrer l'application.

Pour le faire, le présent rapport doit être bien structuré pour être exploité après la mise en place de l'application, et nous l'avons organisé de la manière suivante :

Dans le premier chapitre, nous décrivons, en premier lieu, l'étude de l'existant du projet qui consiste à dévoiler la méthode actuelle utilisée en relevant ses insuffisances et proposant une solution. En deuxième lieu, nous déterminons les principales fonctionnalités offertes par l'application en précisant ses besoins fonctionnels et non fonctionnels. Ensuite, nous présentons le cadre de projet.

Le deuxième chapitre aborde la phase de conception. En premier lieu, nous spécifions la méthodologie adaptée, après, nous présentons les différents diagrammes.

Finalement, au niveau du dernier chapitre intitulé réalisation, nous présentons notre environnement de travail matériel et logiciel, ainsi que les principales interfaces graphiques réalisées.

# *Chapitre 1 : Spécification et Analyse des besoins*

### Introduction

Dans ce chapitre, nous mettons le sujet dans son cadre général. Par la suite, nous abordons l'étude de l'existant du projet, suivie d'une critique pour pouvoir dégager les contraintes à respecter pendant la réalisation de notre projet.

Ainsi, ce chapitre présente l'ensemble des besoins qu'ils soient fonctionnels et non fonctionnels.

## I. Etude de l'existant

### 1. Description de l'existant

Comme tous les établissements de l'enseignement supérieur l'Université Virtuelle de Tunis offre à ces étudiants la possibilité d'effectuer des stages dans des entreprises publiques ou privées.

En effet, après l'achèvement des procédures administratives d'un stage, les étudiants sont affectés à un ou plusieurs encadreurs selon le type de stage.

Les étudiants sont intégrés alors durant la période du stage dans les entreprises d'accueil sous l'assistance de l'encadreur société et aussi l'encadreur de l'université.

Le processus de suivi des stages se fait manuellement. En effet, l'échange d'information et de la documentation entre l'encadreur et l'étudiant se fait par le mailing ou par papiers manuscrits.

L'évaluation finale du projet avant les soutenances est effectuée dans un temps réduit, les membres du jury d'un stage n'ont aucune idée sur l'avancement d'un stage ni l'achèvement des tâches.

### 2. Critique de l'existant

La procédure existante atteint ces objectifs depuis sa mise en place mais le critère « suivi en temps réel » est absent, en effet :

- Le contrôle des étudiants n'est effectué que par une seule partie relative à l'emplacement de l'étudiant durant la période du stage ;
- Manque de collaboration entre l'université et les entreprises ;
- Dans certains cas les encadreurs ne se rencontrent que lors des soutenances ;
- La tâche affectée par les deux encadreurs diffère parfois en termes de finalité ou objectif ;

- L'évaluation finale du projet avant les soutenances est effectuée dans un temps réduit, les membres de jury d'un stage n'ont aucune idée sur l'avancement d'un stage ni l'achèvement des tâches ;
  - Perte de temps au niveau de l'échange des documents ;
  - Risque de perte de document ;
  - Absence de plan d'action du projet validé par les deux encadreurs ;
  - Manque de support académique (Entreprise) ;
  - Absence de réunion hebdomadaire entre tous les intervenants.
- ⇒ Absence d'une plateforme ou il y'a des forums de discussion ...

### 3. Solution proposée

Afin de palier aux défaillances observé, nous proposons d'informatiser le processus de suivi des stages.

Pour cela, nous optons pour le développement d'une application web dynamique qui va être exploité par plusieurs intervenants et son principal objectif est le suivi des stages.

L'application doit pouvoir faciliter le processus de suivi des stages et permet à l'étudiant de communiquer avec ses encadreurs des documents et de saisir leurs activités ou l'avancement réalisé.

L'application permet aussi aux encadreurs et aux membres de jury de suivre et valider le travail réalisé par l'étudiant.

## II. Etude des besoins

Dans cette section du chapitre, nous nous intéressons aux besoins des utilisateurs à travers les spécifications fonctionnelles et non fonctionnelles pour aboutir à une application de qualité selon les besoins du client.

### 1. Besoins fonctionnels

Les besoins fonctionnels se présentent en quatre grandes parties, l'échange de l'information et de la documentation entre l'étudiant et l'encadreur, la gestion du cahier du stage, le forum et la gestion du BackOffice.

***a. L'échange de l'information et de la documentation***

➤ Gestion de la documentation

Notre application doit permettre aux utilisateurs L'échange de l'information et de la documentation en assurant aux utilisateurs l'ajout, la modification et la consultation des documents et aussi la gestion des commentaires sur les documents.

➤ Affectation des tâches

Chaque encadreur durant la période de stage peut affecter plusieurs tâches aux étudiants et spécifier la date limite de la remise du travail demandé.

***b. La gestion du cahier du stage***

La gestion du cahier du stage se traduit par l'ajout des tâches réalisé par les étudiant pour chaque journée, ainsi que la modification et la consultation son contenue.

***c. Le Forum***

Notre application doit intégrer un forum de discussions afin de permettre aux utilisateurs d'échanger l'information entre eux, est cela par l'ajout des sujets de discussion, ainsi que l'ajout des réponses aux sujets ajoutés.

***d. La gestion du BackOffice***

Le BackOffice est la partie ou est assuré le paramétrage de notre application. Cette partie intégrera la gestion des stages, la gestion des encadreurs, la gestion des étudiants, la gestion des membres de jury et la gestion des profils de chaque utilisateur.

La gestion des stages se traduit par l'ajout, la modification et la suppression des stages, ainsi que la gestion des types de stages et l'affectation des membres de jury pour chaque stage.

**2. Besoins non fonctionnels**

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

- Fiabilité : l'application doit fonctionner de façon cohérente sans erreurs ;
- Les erreurs : l'application doit les signalées par des messages d'erreurs ;

- Ergonomie et bon IHM : l'application doit être adaptée à l'utilisateur sans qu'il fournisse trop d'effort (utilisation claire et facile) ;
- Efficacité : l'application doit permettre l'accomplissement de la tâche avec le minimum de manipulations.
- Sécurité : l'application doit être sécurisée au niveau des données: authentification et contrôle d'accès.

### III. Cadre de projet

Dans le cadre de notre projet de fin d'étude au sein de l'Université Virtuelle de Tunis nous avons eu comme tâche de concevoir et développer une plateforme numérique de suivi des stages.

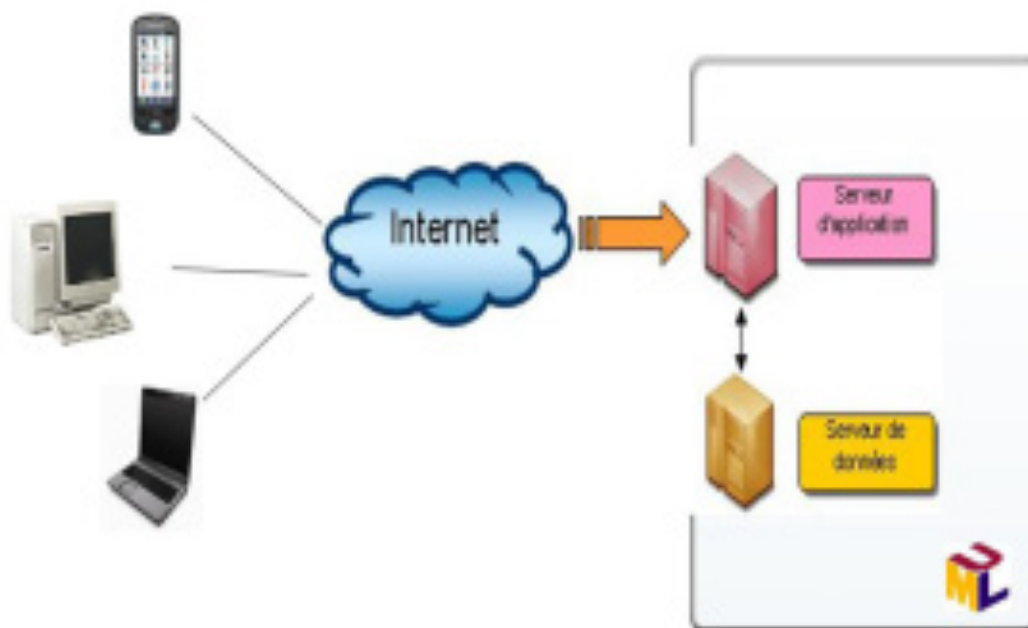


Figure 1: Schéma présentatif du cadre de projet

### Conclusion

Dans ce chapitre on a présenté une étude du système existant, les lacunes qu'il comprend ainsi que la solution proposée pour pallier ces problèmes. Ainsi que les besoins fonctionnels et non fonctionnels qui ont permis de mieux expliciter le système à réaliser.

Dans le chapitre suivant nous entamons l'étude conceptuelle.

## *Chapitre 2 : 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.

## I. Conception Générale

### 1. Le modèle de cycle de vie

Afin de concevoir et réaliser notre application, nous avons opté pour le model de cycle de vie en V.

#### a. Présentation du model

Le modèle du cycle en V est un modèle conceptuel de gestion de projet imaginé suite au problème de réactivité du modèle en cascade. Il permet, en cas d'anomalie, de limiter un retour aux étapes précédentes.

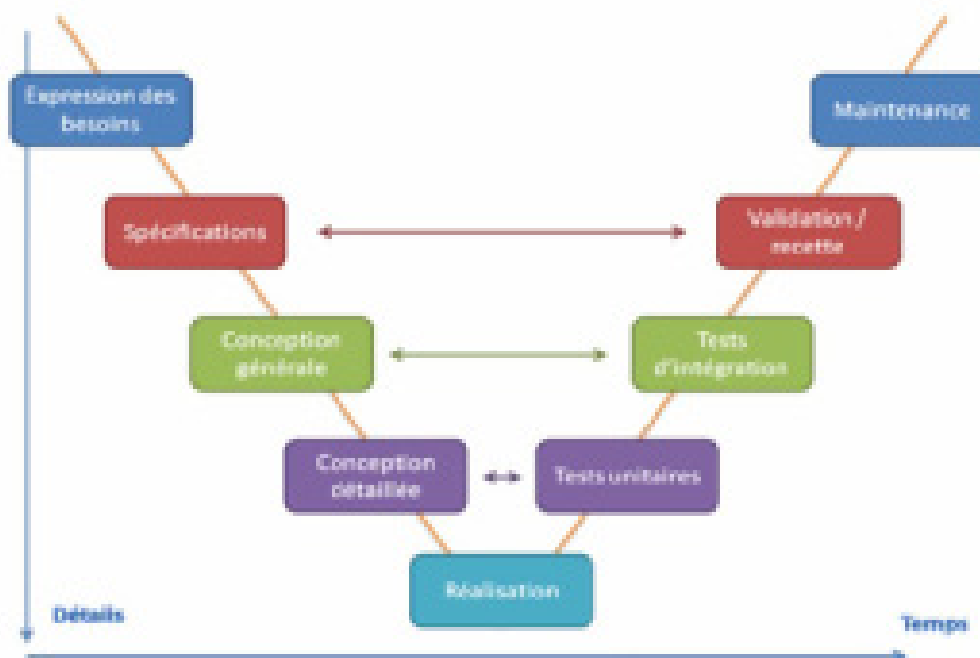


Figure 2: Le model en V

### *b. Description du modèle*

La représentation en V tient d'avantage compte de la réalité, le processus de développement n'est pas réduit à un enchaînement de tâches séquentielles.

Elle montre que:

- c'est en phase de **spécification** que l'on se préoccupe des procédures de **validation** ;
- c'est en phase de **conception générale** que l'on se préoccupe des procédures d'**intégration** ;
- c'est en phase de **conception détaillée** que l'on prépare les **tests unitaires**.

Le modèle de cycle de vie en V permet d'anticiper sur les phases ultérieures de développement du produit. En particulier le modèle en V permet de commencer plus tôt:

- Plan de tests de qualification ;
- Plan d'évaluation des performances.

## 2. Méthodologie adoptée

Afin d'optimiser la compréhension ainsi que la réalisation du projet on a choisi comme méthodologie de développement RUP (Rational Unified Process). C'est un standard qui propose une conception assez poussée et prend en compte tous les détails de conception y compris le système existant, RUP a la visibilité sur tout le système d'information et il est conseillé pour des projets qui commencent à zéro.

Cette méthodologie est basée sur le langage de modélisation UML (Unified Modeling Language) que nous avons adopté pour toute la suite de notre travail.

En effet UML (Langage de modélisation unifié) comme son nom l'implique, un langage de modélisation et non une méthode ou un procédé. L'**UML** est constitué d'une notation très spécifique ainsi que les règles grammaticales s'y attachant pour élaborer des modèles de logiciel.

L'**UML** supporte un riche ensemble d'éléments de notation graphique. Il décrit la notation pour les classes, les composants, les nœuds, les activités, le workflow, les cas d'utilisations, les objets, les états ainsi que la façon de modéliser les relations entre ces éléments. L'**UML** permet également les extensions personnelles à travers les éléments stéréotypés.

### 3. Concept et architecture de l'AGSS (Application de Gestion et Suivi des Stages)

Notre projet consiste à concevoir une application web avec une architecture 3-tiers.

Dans l'architecture à 3 niveaux (appelée architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

- Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation ;
- Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur ;
- Le serveur de données, fournissant au serveur d'application les données dont il a besoin.

Ce modèle d'architecture 3-tiers a pour objectif de répondre aux préoccupations suivantes :

- allègement du poste de travail client ;
- prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.) ;
- introduction de clients dits " légers " (plus liée aux technologies Intranet/HTML qu'au 3-tiers proprement dit) ;
- et enfin, meilleure répartition de la charge entre différents serveurs d'application.

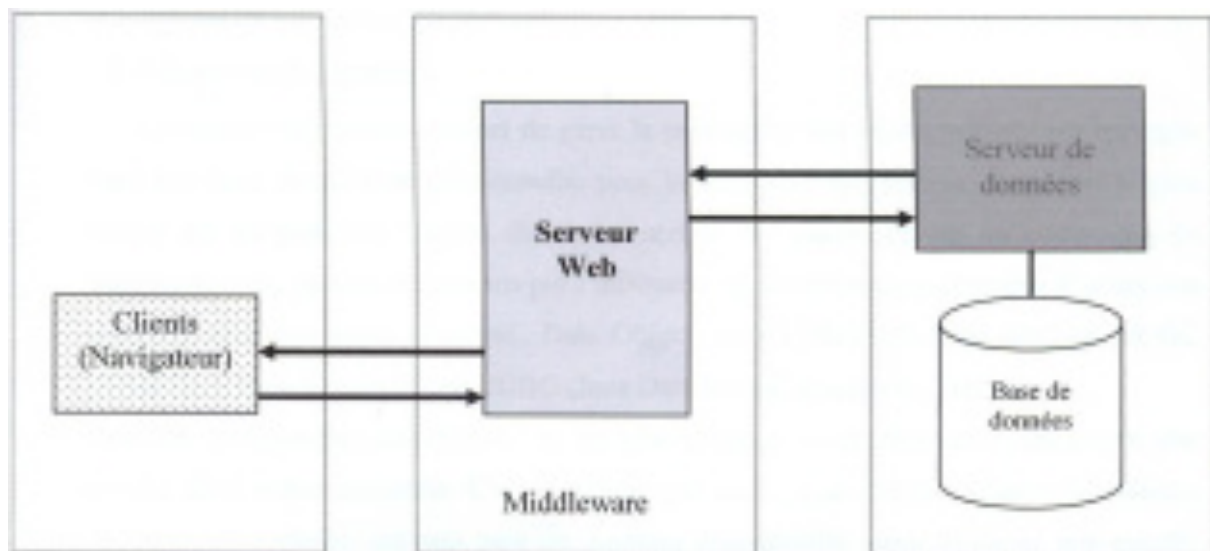


Figure 3: La structure générique d'une architecture 3-tiers

## II. Conception détaillé

### 1. Les diagrammes des 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.

#### *a. 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 application, nous avons défini quatre acteurs qui sont :

#### **L'étudiant**

- Gestion de la documentation
- Gérer le cahier de stage
- Participer au forum
- Ajouter des remarques sur le stage

#### **Les encadreurs**

- Gestion de la documentation
- Affecter des tâches pour les étudiants
- Consulter le cahier de stage
- Participer au forum
- Ajouter des remarques sur le stage

#### **Membre de jury**

- Consulter les documents
- Consulter le cahier du stage
- Consulter les tâches affectées par les encadreurs
- Valider les stages

#### **Administrateur**

- Gestion du BackOffice

- Paramétrer l'application
- Gérer les profils
- Gérer les stages
- Gérer les étudiants
- Gérer les encadreurs
- Gérer les membres de jury

### ***b. Description des cas d'utilisation***

L'étude des cas d'utilisation a pour objectif de déterminer ce que chaque utilisateur attend du système. La détermination des besoins est basée sur la représentation de l'interaction entre l'acteur et le système.

- *Diagramme de cas d'utilisation de l'Etudiant*

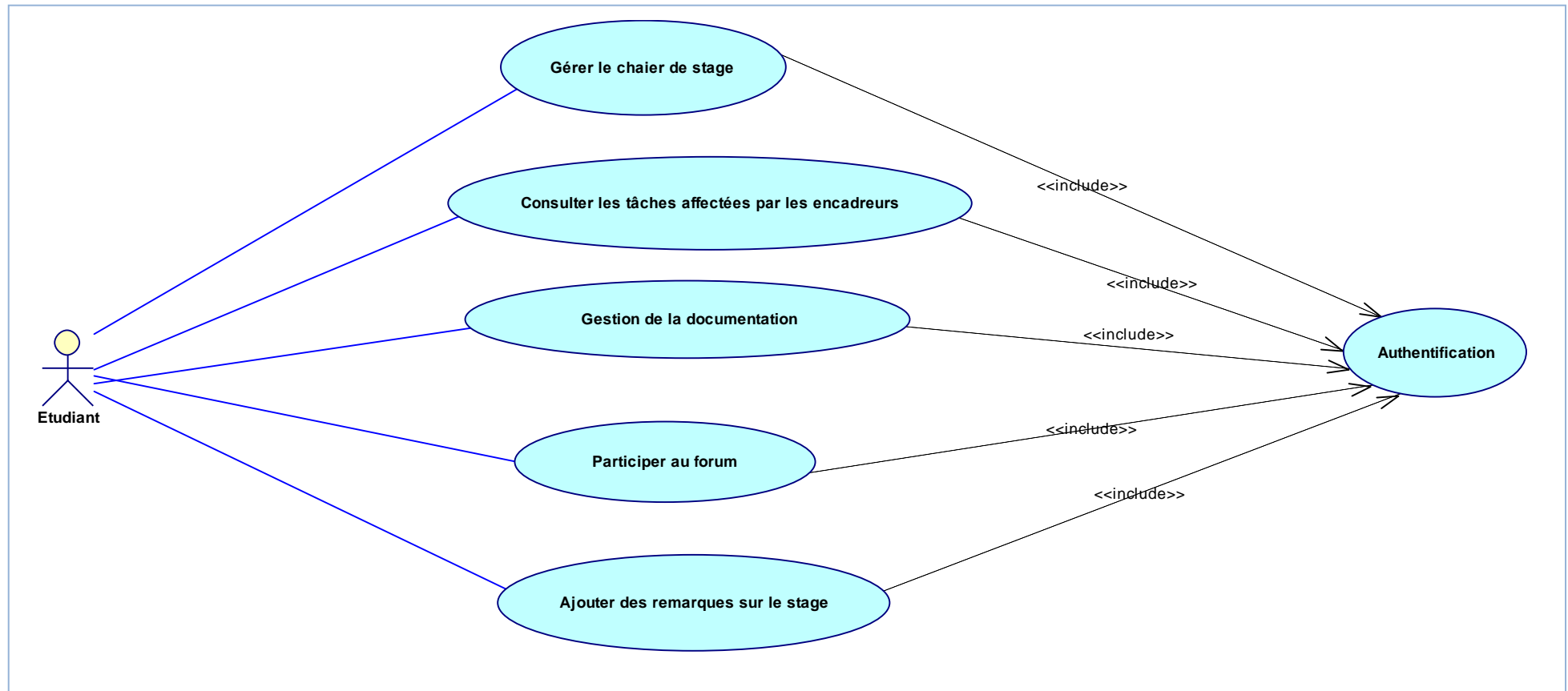
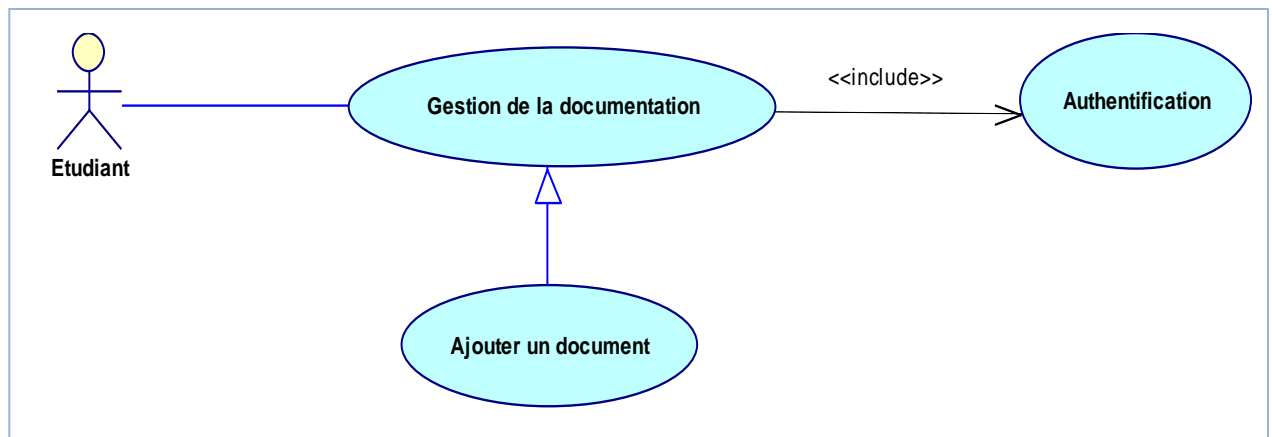
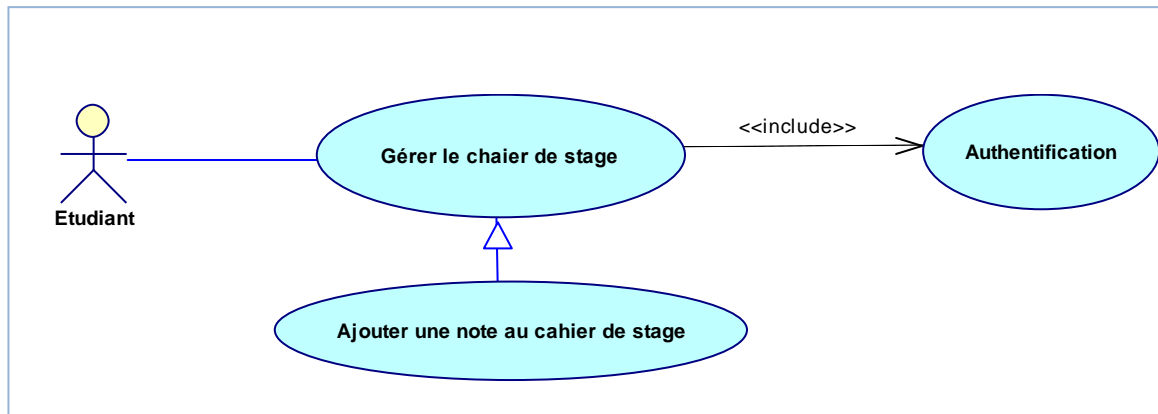


Figure 4: Diagramme de cas d'utilisation de l'étudiant



➤ *Description du cas d'utilisation « Ajouter un document »*

SOMMAIRE D'IDENTIFICATION	
<b>Titre :</b>	Ajouter un document
<b>But :</b>	L'échange de documentation entre l'étudiant et les autres acteurs
<b>Résumé :</b>	L'étudiant choisit le document puis valide l'action. Le système uploade le document sur le serveur et ajoute le document à la base de données.
<b>Acteur:</b>	Etudiant
DESCRIPTION DES ENCHAINEMENTS	
Pré condition	Post condition
- L'étudiant doit être authentifié	- Le document est déposé
Enchaînement nominal	
<ul style="list-style-type: none"> <li>- L'étudiant parcourt le document</li> <li>- L'étudiant valide l'action</li> <li>- Le système vérifie le type de document</li> <li>- Le système uploade le document sur le serveur</li> <li>- Le système met à jour la base de données</li> </ul>	



➤ *Description du cas d'utilisation « Ajouter des notes au cahier de stage »*

SOMMAIRE D'IDENTIFICATION	
<b>Titre :</b>	Ajouter des notes au cahier de stage
<b>But :</b>	Le remplissage du cahier de stage
<b>Résumé :</b>	L'étudiant remplit le formulaire du cahier de stage puis valide l'action. Le système met à jours le cahier de stage.
<b>Acteur:</b>	Etudiant
DESCRIPTION DES ENCHAINEMENT	
Pré condition	Post condition
- L'étudiant doit être authentifié	- Les données sont enregistrées
Enchaînement nominal	
<ul style="list-style-type: none"> <li>- L'étudiant Choisit la date</li> <li>- L'étudiant remplit le formulaire</li> <li>- L'étudiant valide l'action</li> <li>- Le système met à jours le cahier de stage</li> </ul>	



- *Diagramme de cas d'utilisation de l'Encadreur*

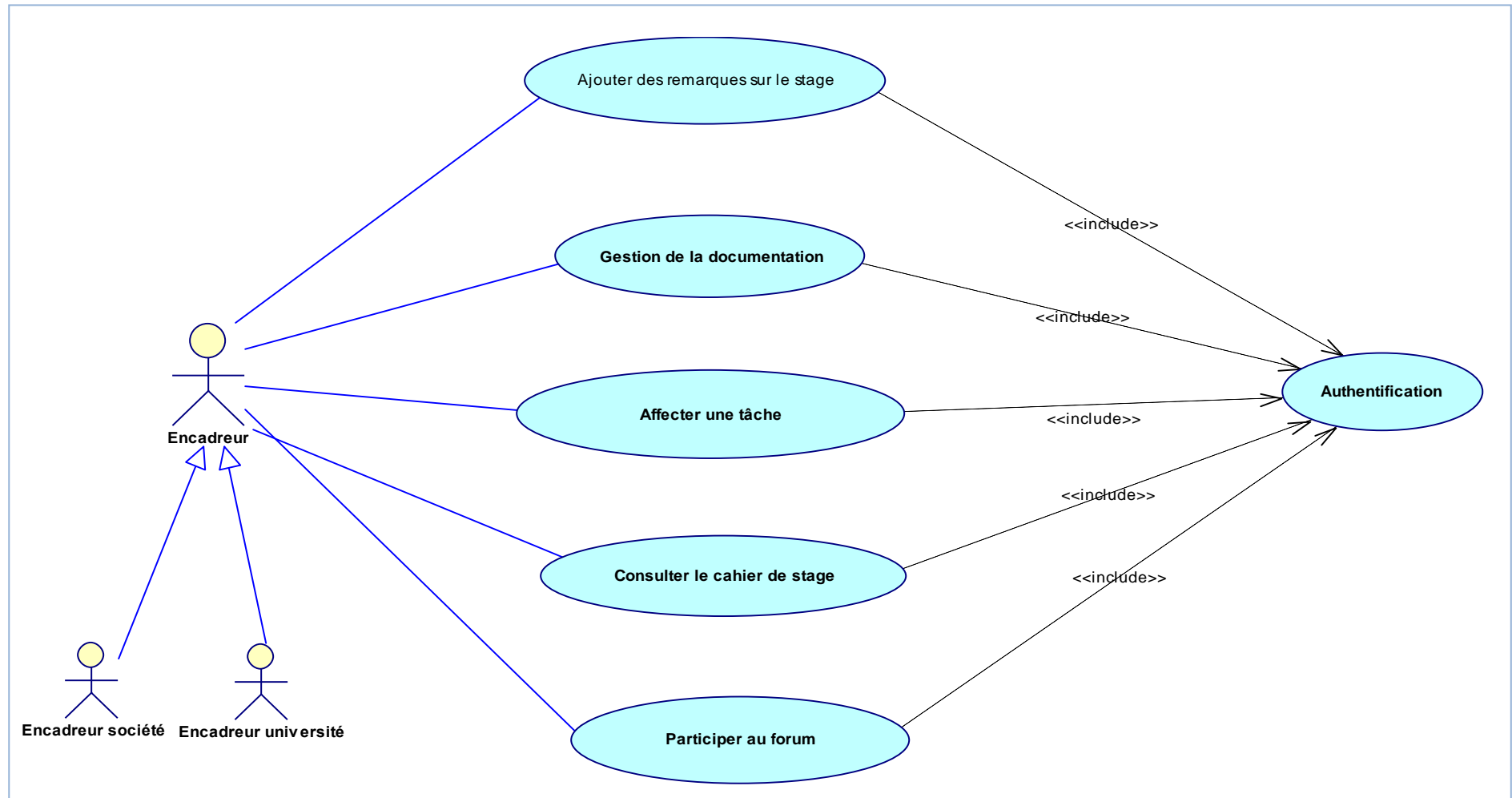
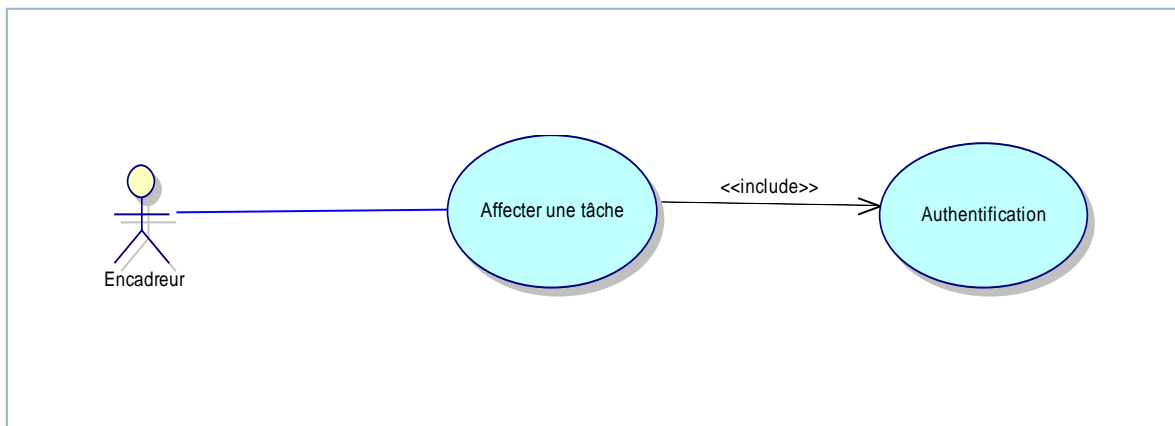


Figure 5: Diagramme de cas d'utilisation de l'encadreur



➤ *Description du cas d'utilisation « Affecter une tâche »*

SOMMAIRE D'IDENTIFICATION	
<b>Titre :</b>	Affecter une tâche
<b>But :</b>	Assister les étudiants pour bien passé leur projet de fin d'étude
<b>Résumé :</b>	L'encadreur remplit le formulaire d'affectation de tâche et valide l'action. Le système enregistre la tâche dans la base de données.
<b>Acteur:</b>	Encadreur
DESCRIPTION DES ENCHAINEMENT	
Pré condition	Post condition
- L'encadreur doit être authentifié	- La tâche est affectée
Enchaînement nominal	
<ul style="list-style-type: none"> <li>- L'encadreur remplit le formulaire</li> <li>- L'encadreur valide l'affectation</li> <li>- Le système enregistre la tâche dans la base de données</li> </ul>	

• *Diagramme de cas d'utilisation du membre de Jury*

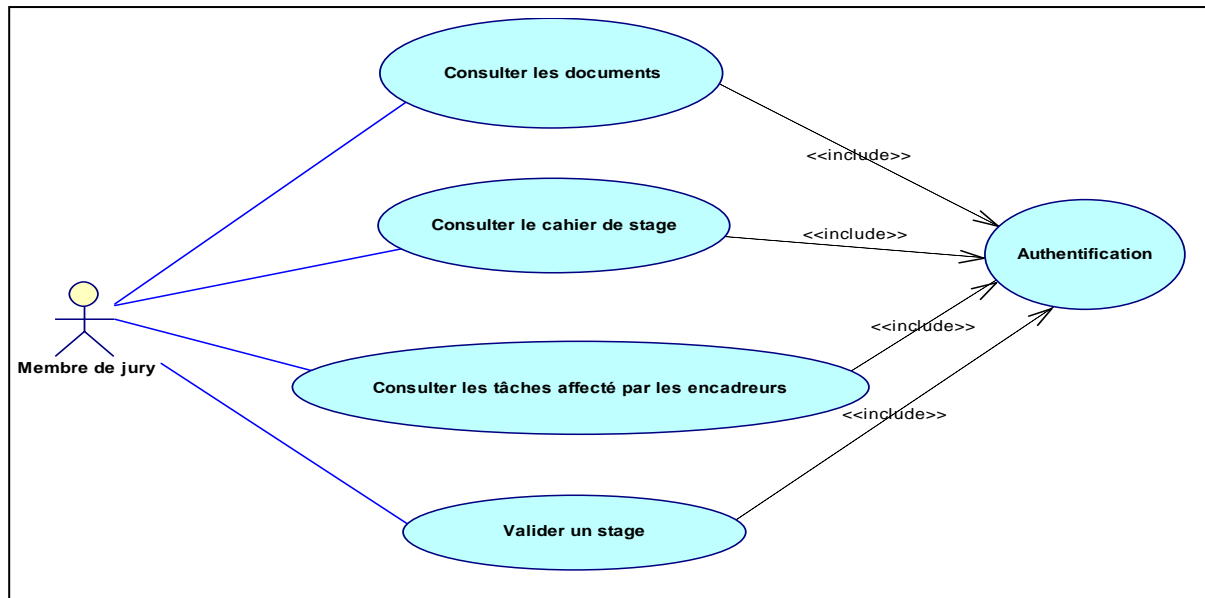


Figure 6: Diagramme de cas d'utilisation du membre de jury

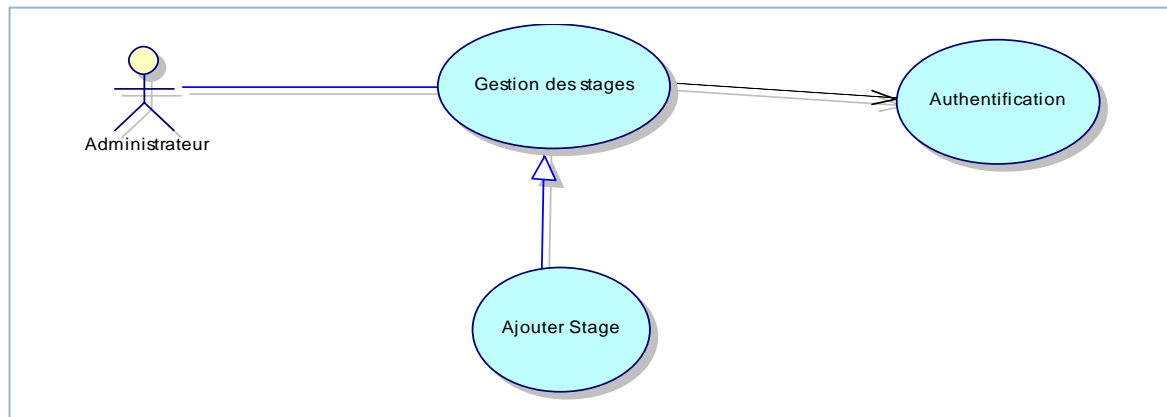
➤ *Description du cas d'utilisation « Valider un stage »*

SOMMAIRE D'IDENTIFICATION	
<b>Titre :</b>	Valider un stage
<b>But :</b>	Valider les stages des étudiants
<b>Résumé :</b>	Le Membre de jury remplit le formulaire de validation et valide l'action. Le système enregistre le statut stage dans la base de données.
<b>Acteur:</b>	Membre de Jury
DESCRIPTION DES ENCHAINEMENT	
Pré condition	Post condition
- Le Membre de Jury doit être authentifié	- Le statut du stage est modifié
Enchaînement nominal	
- Le Membre de Jury remplit le formulaire - Le Membre de Jury valide l'action - Le Système enregistre le statu du stage dans la base des données	

- *Diagramme de cas d'utilisation relatif au BackOffice*



Figure 7: Diagramme de cas d'utilisation de l'Administrateur



➤ Description du cas d'utilisation « Ajouter un stage »

SOMMAIRE D'IDENTIFICATION	
<b>Titre :</b>	Ajouter un stage
<b>But :</b>	Ajouter les stages des étudiants
<b>Résumé :</b>	L'administrateur remplit le formulaire d'ajout de stage et valide l'action. Le système enregistre le stage dans la base de données.
<b>Acteur:</b>	Administrateur
DESCRIPTION DES ENCHAINEMENTS	
Pré condition	Post condition
- L'administrateur doit être authentifié	- Le stage est ajouté
Enchaînement nominal	
- L'administrateur remplit le formulaire - L'administrateur valide l'ajout - Le système enregistre le stage dans la base de données	

## 2. Les diagrammes d'activité

Le **diagramme d'activité** est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (Multi-threads ou Multi-processus).

Il permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Il est recommandable pour exprimer une dimension temporelle sur une partie du modèle.

### ➤ Front Office

#### a. Diagramme d'activité Affectation des tâches par l'encadreur

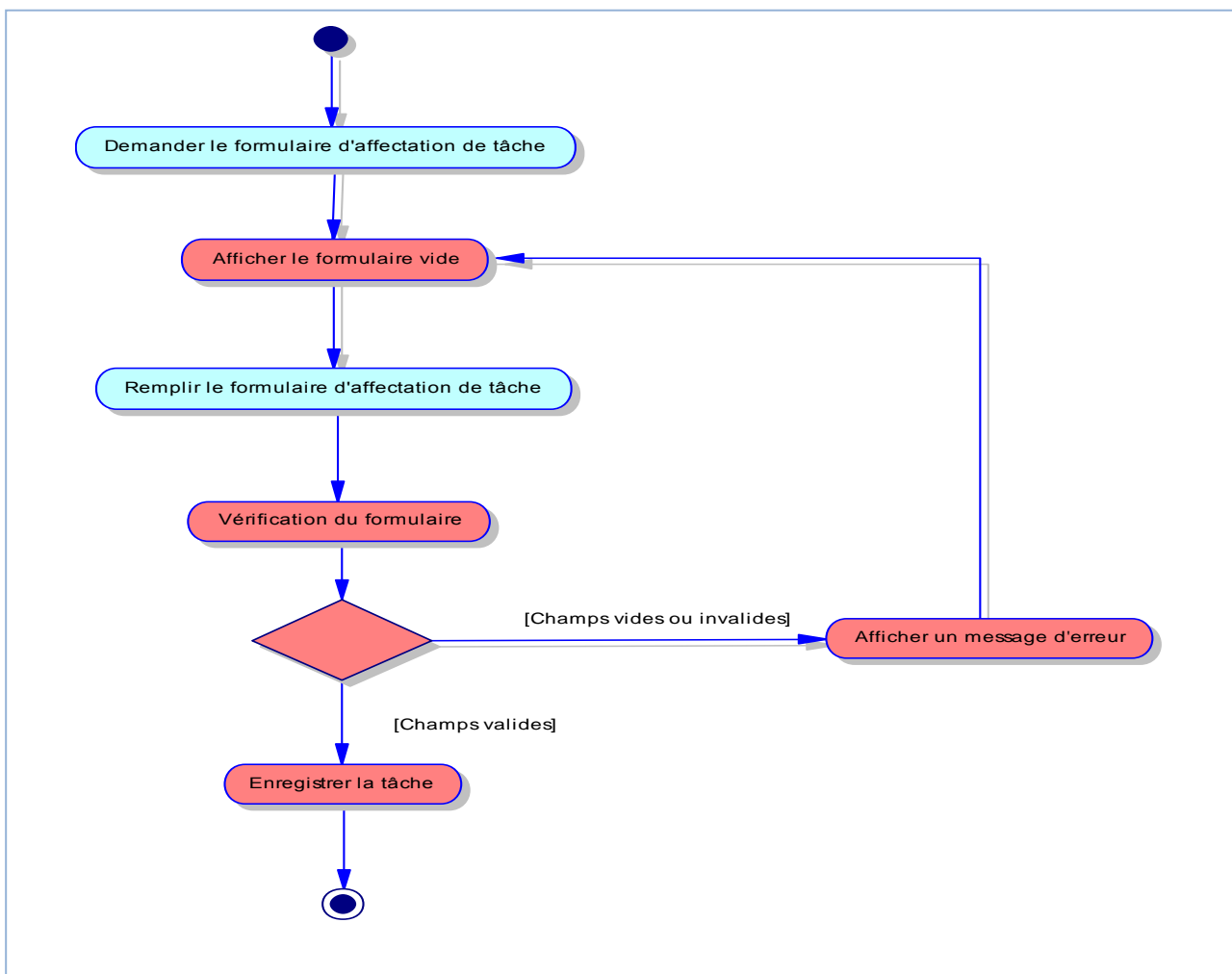


Figure 8: Diagramme d'activité Affectation des tâches par l'encadreur

*b. Diagramme d'activité Remplir le cahier de stage*

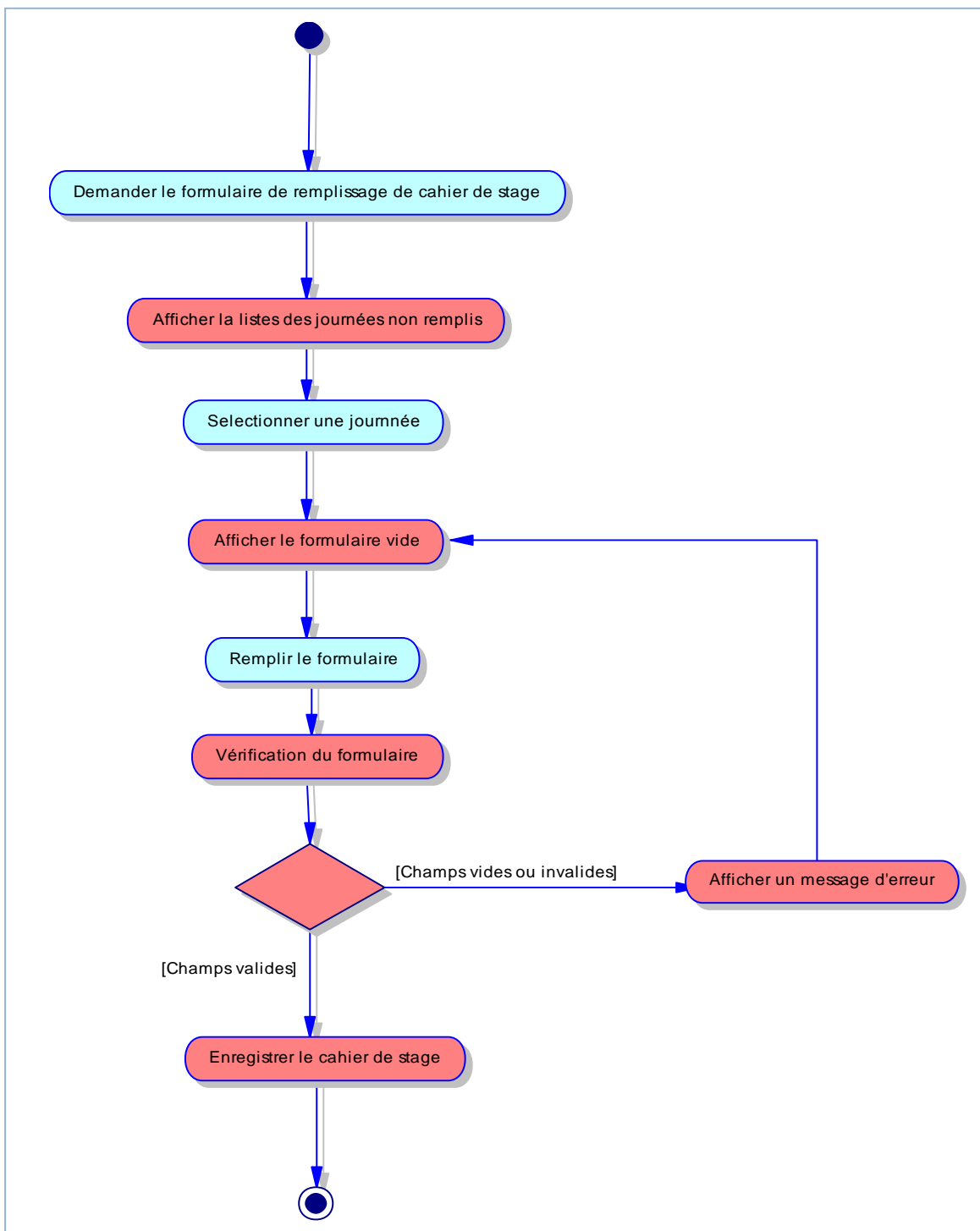


Figure 9: Diagramme d'activité Remplir le cahier de stage

## ➤ Back Office

### c. Diagramme d'activité Ajouter un stage

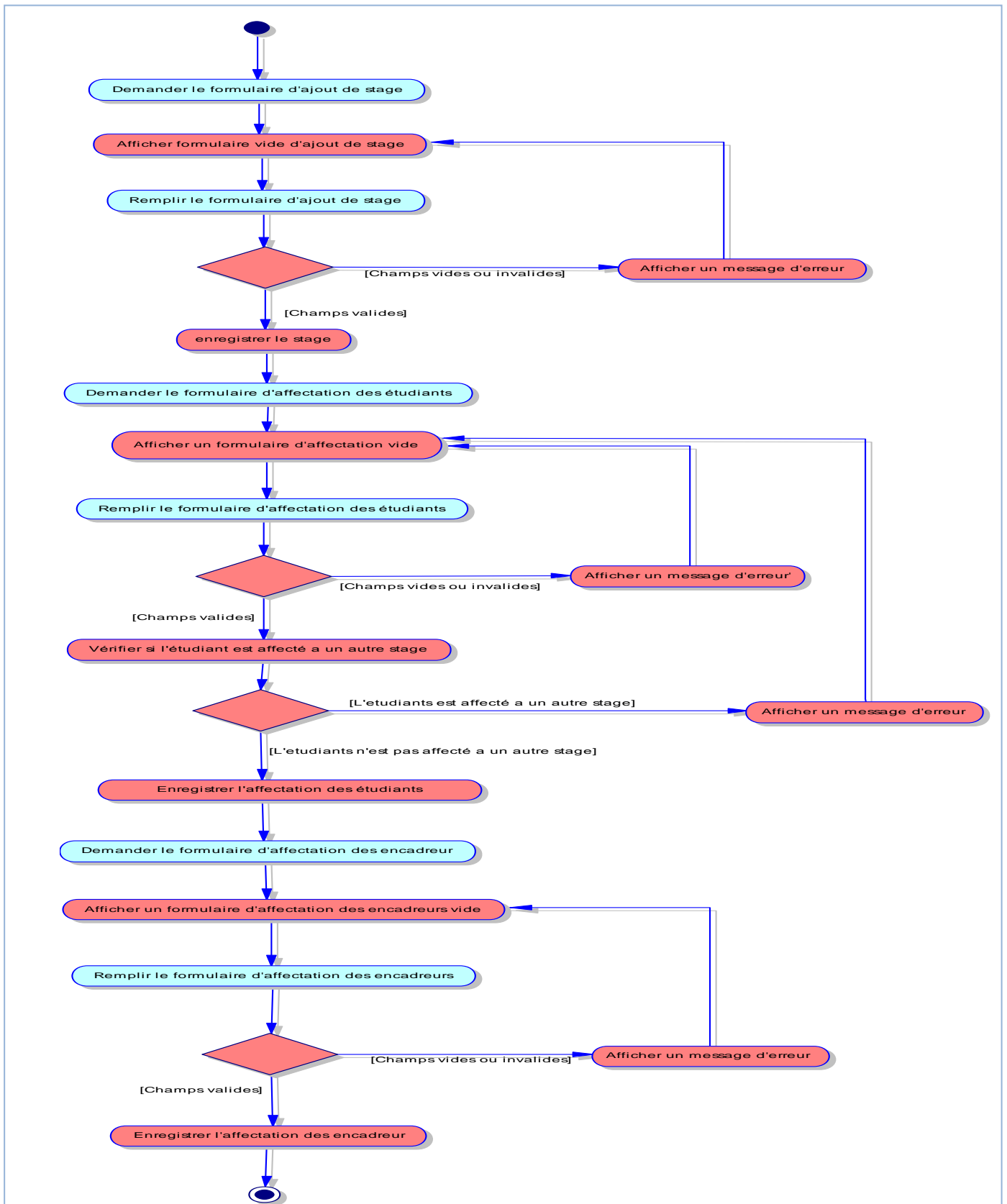


Figure 10: Diagramme d'activité Ajouter un stage



### 3. Les diagrammes de séquences

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.

#### a. Diagramme de séquence d'authentification

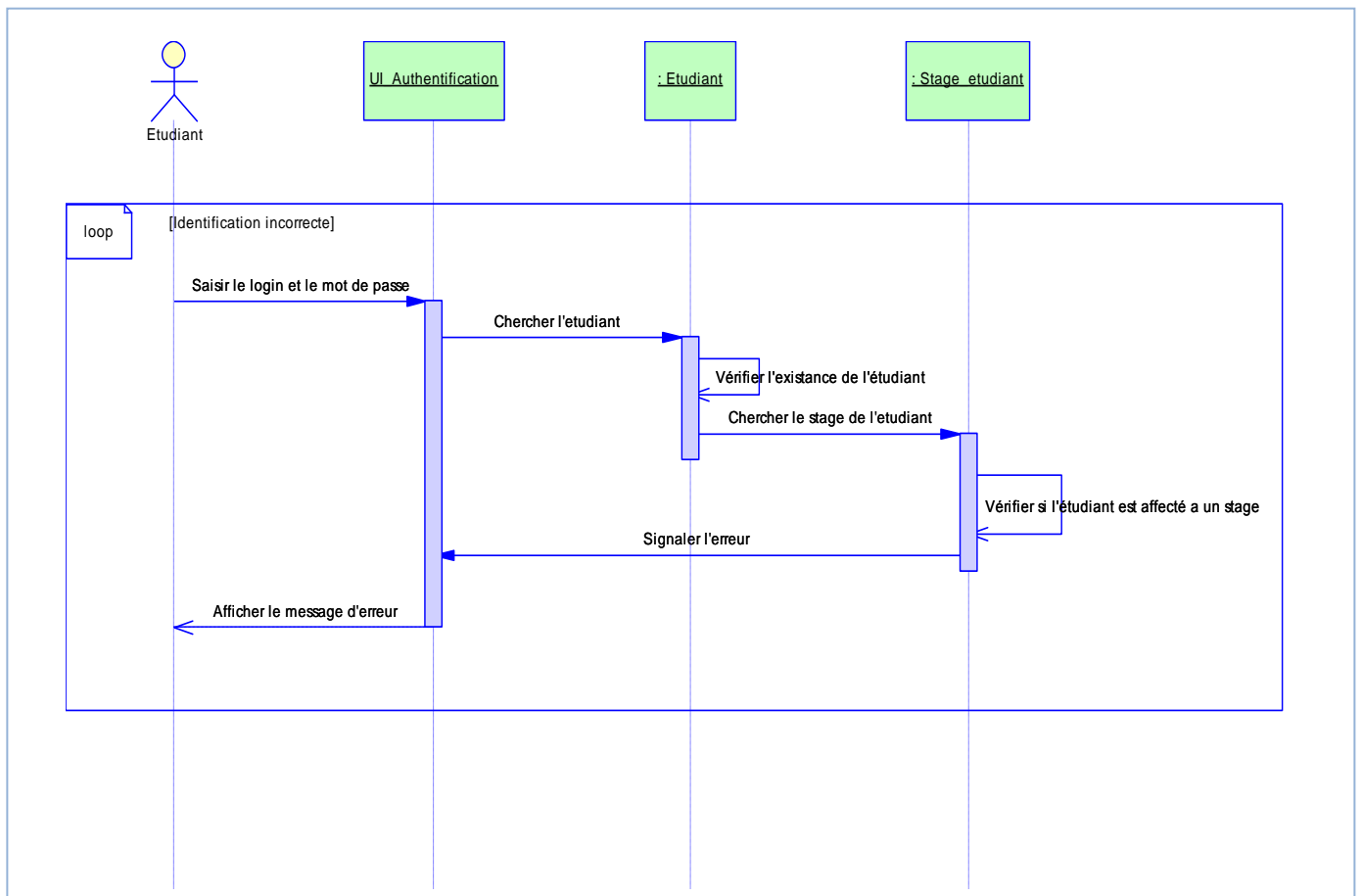


Figure 11: Diagramme de séquence d'authentification

*b. Diagramme de séquence déposer un document*

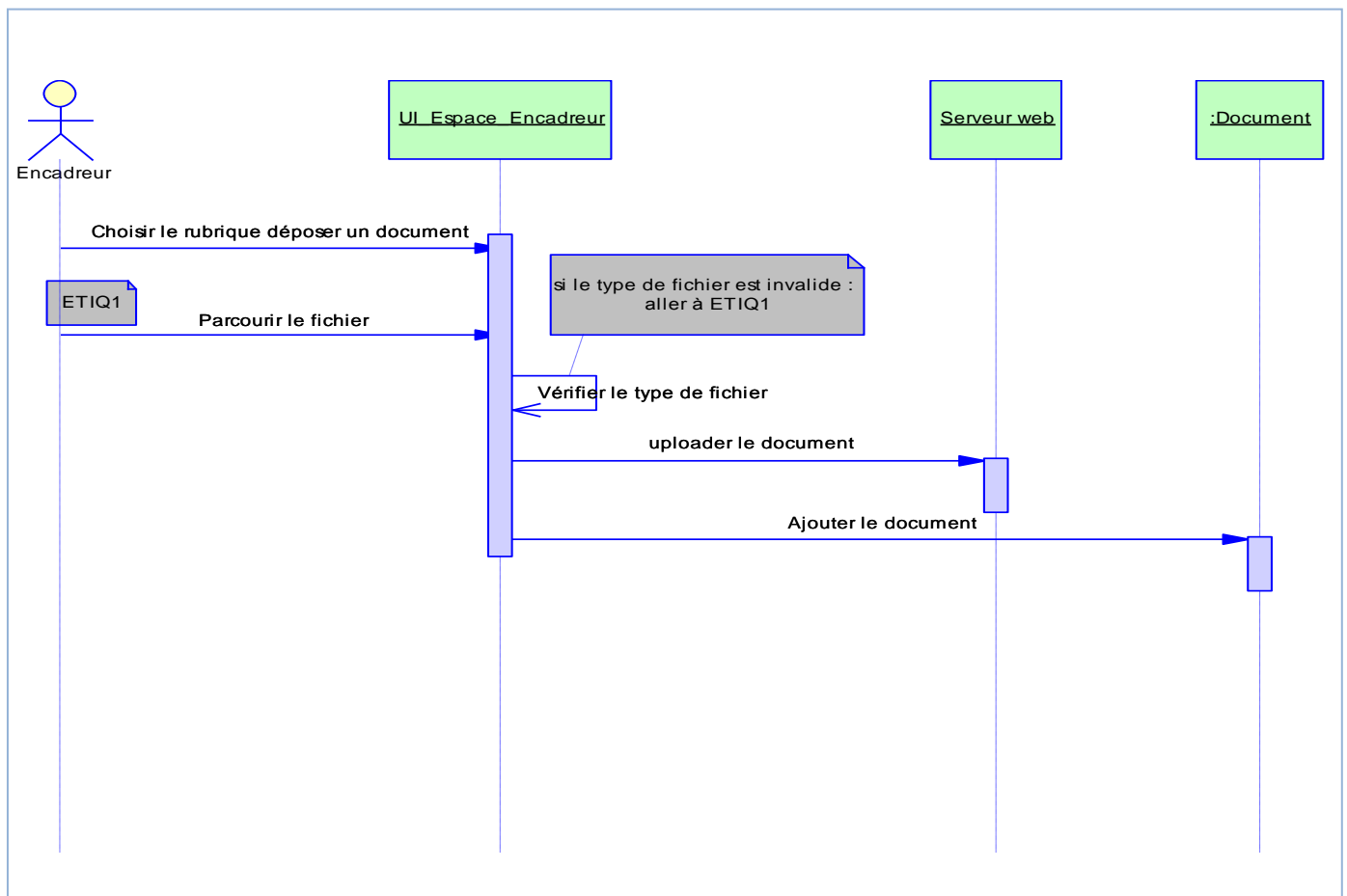


Figure 12: Diagramme de séquence déposer un document

*c. Diagramme de séquence valider un stage*

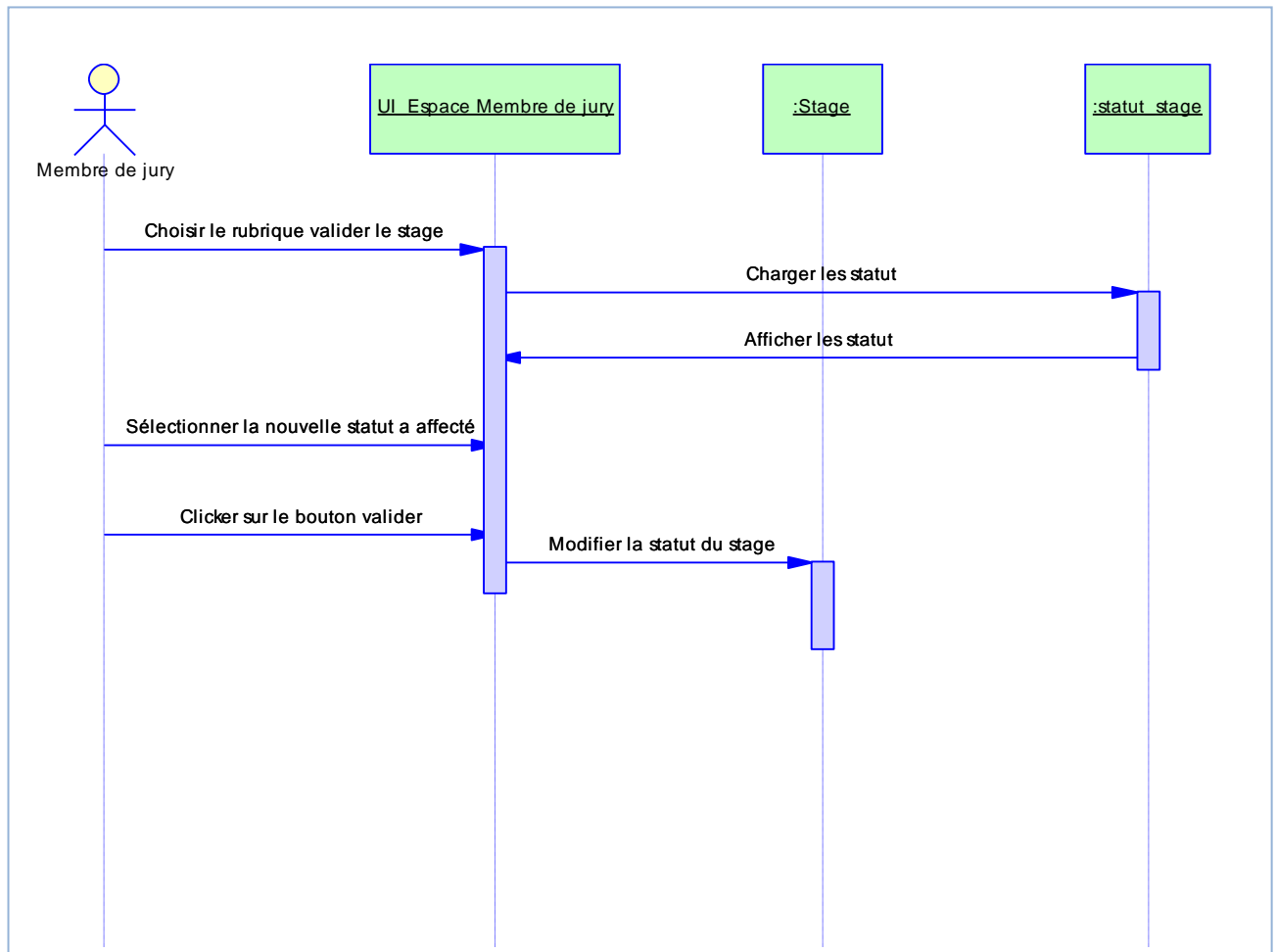


Figure 13: Diagramme de séquence valider un stage

d. Diagramme de séquence ajouter un stage

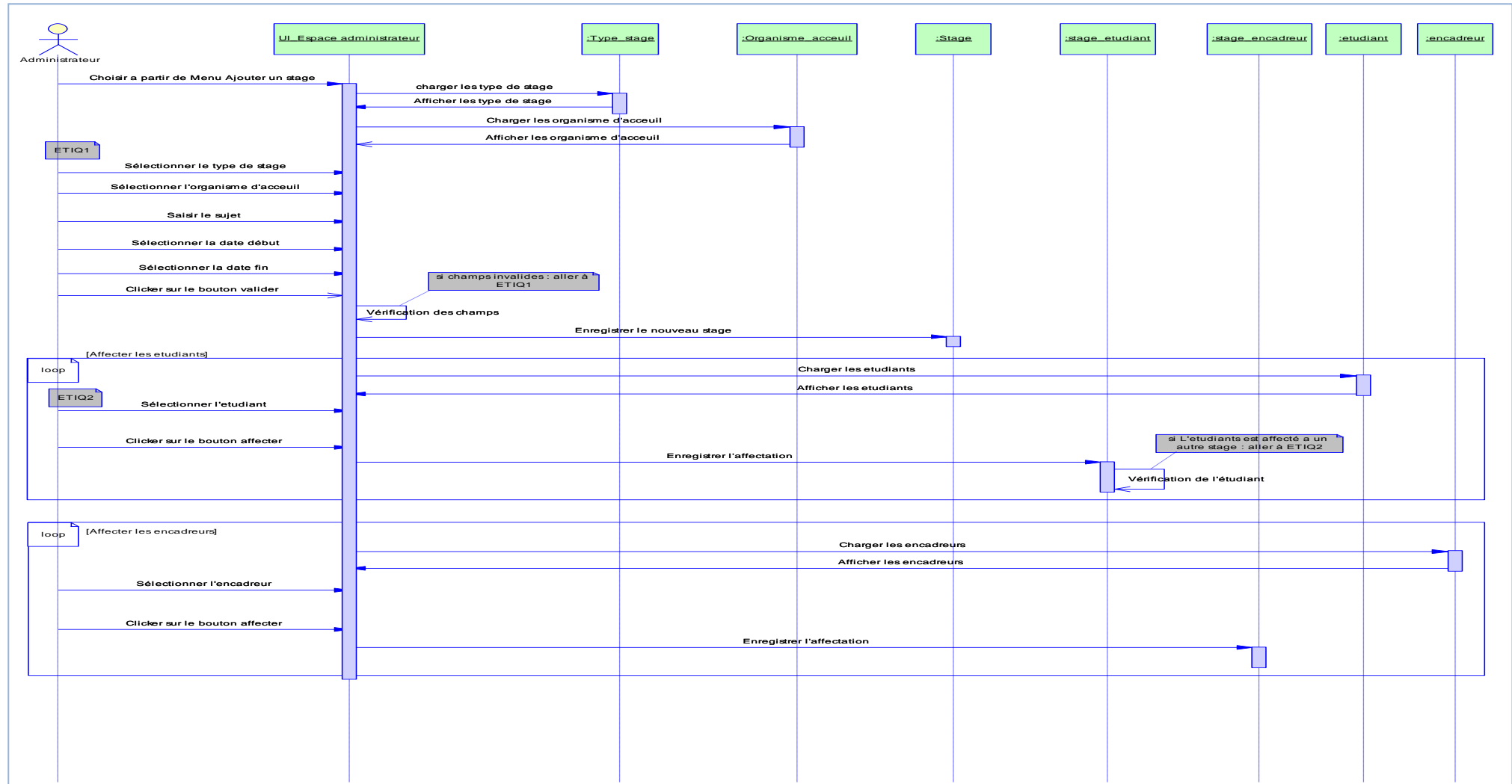


Figure 14: Diagramme de séquence ajouter un stage

### 4. Diagramme des classes

#### *a. Description des classes*

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.

⇒ Notre application comporte les classes suivantes :

**Classe Stage** : elle contient toutes les informations d'un stage.

**Classe Etudiant** : elle contient toutes les informations d'un étudiant.

**Classe Encadreur** : elle contient toutes les informations d'un encadreur.

**Classe Membre\_jury** : elle contient toutes les informations d'un membre de jury.

**Classe Encadreur\_stage** : elle indique pour chaque stage les encadreurs affectés et le type de l'encadreur.

**Classe Etudiant\_stage** : elle indique pour chaque stage les étudiants affectés.

**Classe Stage\_jury** : indique pour chaque stage les membres de jury de la soutenance.

**Classe Profils\_encadreur** : contient les paramètres d'accès de chaque encadreur.

**Classe Profils\_jury** : contient les paramètres d'accès de chaque membre de jury.

**Classe Pays** : contient les différents pays.

**Classe ville** : contient les différentes villes d'un pays.

**Classe Cahier\_stage** : contient les différentes tâches effectuées au cours de la durée d'un stage.

**Classe Tache** : elle indique pour chaque stage les différentes tâches affectées par les encadreurs.

**Classe Type\_stage** : contient les différents types d'un stage.

**Classe Document** : contient pour chaque stage les différents document uploadé.

**Classe Commentaire\_document** : elle contient tous les commentaires d'un document.

**Classe Organisme\_Accueil** : contient les différents organismes d'accueil.

**Classe organisme\_service\_type** : elle comporte les différents types des organismes.

**Classe Stage\_statut** : Contient les statuts d'un stage.

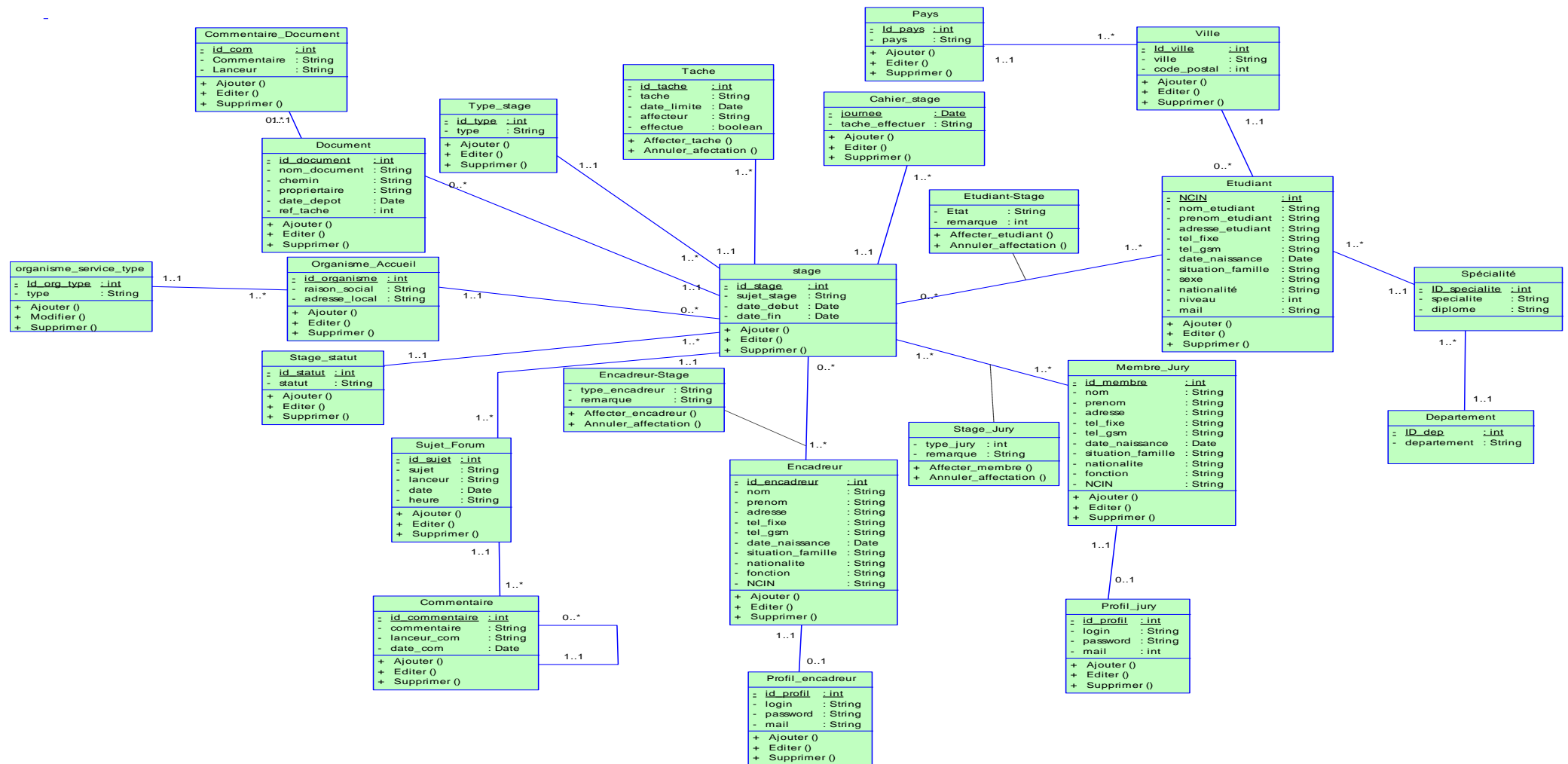
**Classe Sujet\_Forum** : elle indique pour chaque stage les différents sujets ajouté.

**Classe commentaire** : elle indique pour chaque sujet de forum les réponses ajoutées.

**Classe spécialité** : elle contient tous les spécialités de l'université

**Classe département** : elle contient tous les départements de l'université.

### *b. Le diagramme des classes*



**Figure 15: Diagramme des classes**

### *c. Classes et attributs*

**Stage** (Id\_stage , Sujet\_stage , date\_debut , date\_fin ,  
#Id\_organisme,#Id\_type,#Id\_validation)

**Etudiant** (NCIN , nom\_etudiant , prenom\_etudiant , adresse\_etudiant , tel\_fixe , tel\_gsm ,  
date\_naissance , situation\_famille , sexe ,nationalite ,niveau ,mail,#Id\_ville, #Id\_specialite)

**Stage\_etudiant** (#NCIN , #Id\_stage , etat,remarque)

**Encadreur**(Id\_encadreur , nom , prenom , adresse, tel\_fixe , tel\_gsm , date\_naissance ,  
situation\_famille , nationalite ,fonction,NCIN)

**Stage\_encadreur**(#Id\_encadreur , #Id\_stage , type\_encadreur,remarque)

**Membre\_jury**(Id\_membre , nom , prenom , adresse, tel\_fixe , tel\_gsm , date\_naissance ,  
situation\_famille , nationalite ,fonction,NCIN)

**Stage\_jury**(#Id\_membre , #Id\_stage , type\_jury, remarque)

**Profil\_encadreur**(Id\_profil , login , password , mail , #Id\_encadreur)

**Profil\_jury**(Id\_profil , login , password , mail , #Id\_jury)

**Cahier\_stage**(Journee ,tache\_effectué , #Id\_stage)

**Tache**(Id\_tache , tache , date\_limite , affecteur , effectue, #Id\_stage)

**Type\_stage**(Id\_type , type)

**Document** (Id\_document , nom\_document , chemin , propriétaire , date\_depot , ref\_tache,  
#Id\_stage)

**Commentaire\_doc**(Id\_com , commentaire , lanceur , #Id\_document)

**Organisme\_acceuil**(Id\_organisme , raison\_social , adresse\_local , #Id\_org\_type)

**Validation** (Id\_validation , type\_validation)

**Sujet\_forum** (Id\_sujet , sujet , lanceur , date , heure , #Id\_stage)

**Commentaire** (Id\_commentaire , commentaire , lanceur , date , heure , #Id\_sujet  
,#Id\_commentaire)

**Ville** (Id\_ville , Ville , #Id\_pays)

**Pays** (Id\_pays , Pays)

**Organisme\_service\_type** (Id\_org\_type , Type)

**Specialite** (ID\_specialite,specialite,diploma,#ID\_dep)

**Departement** (ID\_dep,department)



### III. Maquette

#### 1. Structure de l'application

Dans notre application, nous choisissons la structure en évolution car elle s'adapte le plus aux besoins de l'utilisateur. L'objectif est de maintenir une hiérarchisation équilibrée qui permet l'accès rapide à l'information et une compréhension intuitive de la façon dont les pages sont organisées tout en donnant la possibilité d'évoluer.

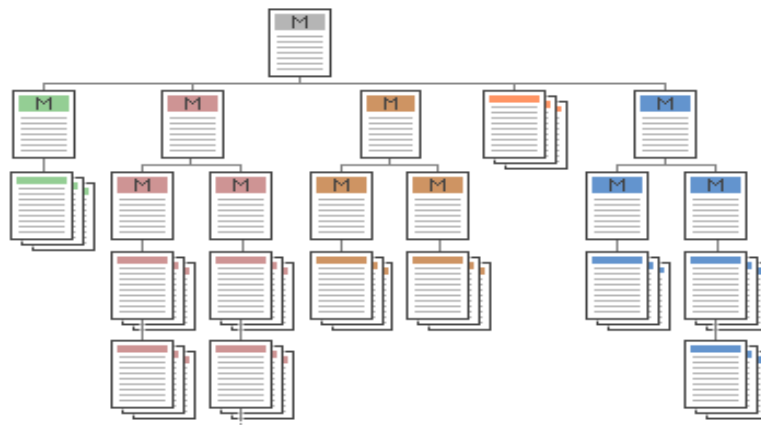


Figure 16: Structure du site en évolution

#### 2. Charte graphique

Le but de la charte graphique est de conserver une cohérence graphique dans les réalisations graphiques d'un même projet pour cela, nous avons choisi la structure suivante pour toutes les interfaces de notre application :

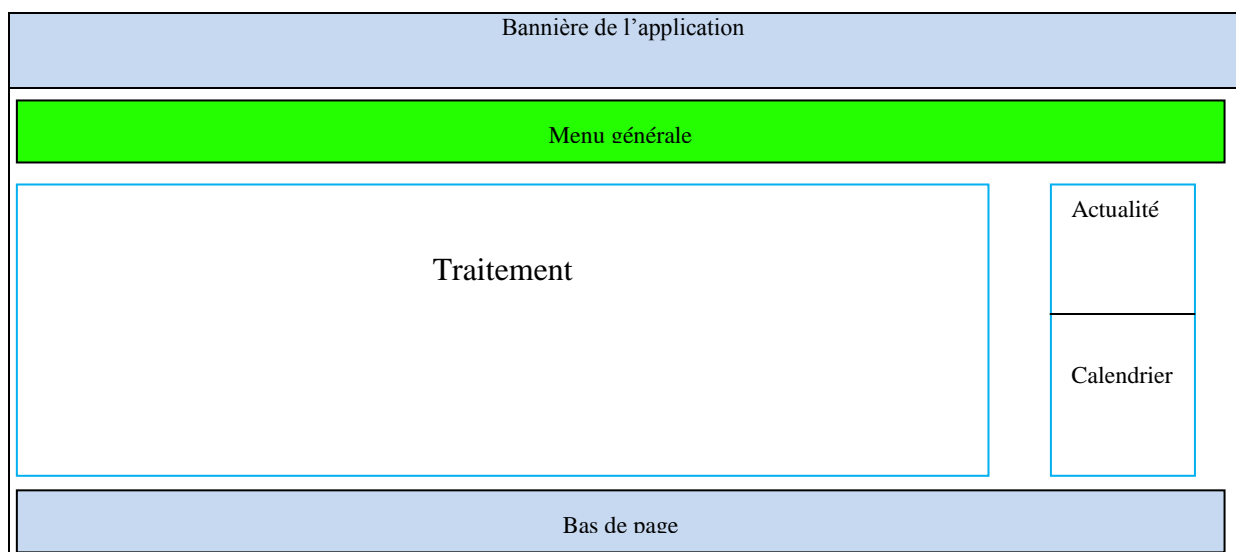


Figure 17: Charte graphique

### Conclusion

Ce chapitre a été consacré à la modélisation de l'aspect statique du système, en se basant sur les spécifications détaillées aux chapitres précédents.

Dans le prochain chapitre on aborde la partie réalisation ainsi que la description de quelques choix techniques effectués pendant le développement.

## *Chapitre 3 : Réalisation*

### Introduction

Au niveau de cette dernière partie, nous nous intéressons aux outils utilisés pour la réalisation de notre application ainsi qu'aux principales interfaces de l'application.

#### I. Environnement du travail

##### 1. Environnement matériel

- PC portable TOSHIBA
- Processeur : Intel Pentium Dual-Core CPU T4300 @ 2.10GHz \* 2
- Mémoire installé : 3 GB

##### 2. Atelier de génie logiciel

###### *a. Langage de programmation*

Nous avons eu recours, pour le développement de notre application au langage de programmation JAVA. C'est la garantie de portabilité qui fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes.

D'autre part JAVA est sécurisée, il a été conçu pour être exploité dans des environnements serveur et distribués. Dans ce cadre, la sécurité n'a pas été négligeable. C'est le langage le plus adopté par les développeurs grâce à sa fiabilité et sa performance élevée.

###### *b. Environnement de développement*

- **JDK 1.6**



Java Développeur Kit Java est l'environnement dans lequel le code Java est compilé pour être transformé en bytecode afin que la machine virtuelle JAVA (JVM) puisse l'interpréter.

Les composants primaires du JDK sont une sélection d'outils de programmation, incluant :

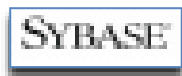
- javac – le compilateur, qui convertit le code source en fichier .class (contenant le bytecode Java) ;
- jar – l'archiveur, qui met sous forme d'un paquetage unique l'ensemble des fichiers class en un fichier JAR ;
- javadoc– le générateur de documentation, qui génère automatiquement de la documentation à partir des commentaires du code source ;
- jdb – le débogueur.



- **JEE**

**Java Enterprise Edition**, ou **Java EE**, est une spécification pour la technique Java de Sun plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au *Framework* Java standard (JSE, *Java Standard Edition*) afin de faciliter la création d'applications réparties.

### *c. Conception*



- **PowerAMC :**

C'est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par SDP sous le nom AMC\*Designor, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque PowerDesigner.

PowerAMC permet de réaliser tous les types de modèles informatiques. Il reste un des seuls qui permet de travailler avec la méthode Merise, cela permet d'améliorer la modélisation, les processus, le coût et la production d'applications.

*d. IDE (Environnement de développement intégré)*



- **MYECLIPSE**

MyEclipse est un IDE construit sur la plate-forme Eclipse, et intègre à la fois des solutions propriétaires et open source dans l'environnement de développement.

MyEclipse a deux versions de base: un professionnel et une édition standard L'édition standard ajoute des outils de base de données, un web designer visuelle, outils de persistance, les outils Spring, Struts et JSF outillage, et un certain nombre d'autres caractéristiques le profil de base Eclipse Java Developer.

*e. Serveur d'Application*



- **TOMCAT**

**Apache Tomcat** est un conteneur libre de servlets Java 2 Enterprise Edition. Tomcat est un projet principal de la fondation Apache. Tomcat implémente les spécifications des servlets et des JSP. Il est paramétrable par des fichiers XML et de propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

*f. Système de gestion de base des données*



- **MySQL**

**MySQL** est un système de gestion de base de données (SGBD). Selon le type d'application. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels.

*g. Framework*

- **HIBERNATE**



**Hibernate** est un framework open source gérant la persistance des objets en base de données relationnelle.

Hibernate est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement J2EE complet.

Hibernate apporte une solution aux problèmes d'adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau.

- **STRUTS**



**Apache Struts** est un framework libre servant au développement d'applications web J2EE. Il utilise et étend l'API Servlet Java afin d'adopter l'architecture Modèle-Vue-Contrôleur.

Cette infrastructure permet la conception et l'implémentation d'applications Web de taille importante par différents groupes de personnes. En d'autres termes, les designers, développeurs de composants logiciels peuvent gérer leur propre part du projet de manière découplée.

Struts permet la structuration d'une application Java sous forme d'un ensemble d'actions représentant des événements déclenchés par les utilisateurs de l'application. Ces actions sont décrites dans un fichier de configuration de type XML décrivant les cheminements possibles entre les différentes actions

#### *h. Design et MultiMedia*



- **HTML**

L'*Hypertext Markup Language*, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables tels que des applets.



- **CSS**

**CSS** (*Cascading Style Sheets* : **feuilles de style en cascade**) est un langage informatique qui sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.



- **JSP**

Le **JavaServer Pages** ou **JSP** est une technique basée sur Java qui permet aux développeurs de générer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique. Depuis la version 2.0 des spécifications, la syntaxe JSP est complètement XML.

La syntaxe du JSP ajoute des balises XML, appelées *actions JSP*, qui peuvent être utilisées pour appeler des fonctions. De plus, cette technique permet la création de bibliothèques de balises JSP (*taglib*) qui agit comme des extensions au HTML ou au XML. Les bibliothèques de balises offrent une méthode indépendante de la plate-forme pour étendre les fonctionnalités d'un serveur HTTP.



## II. Scénarios Applicatif

Cette partie recense la présentation d'un Scénario applicatif de l'application, vu que la navigation dans l'application dépend de l'utilisateur, nous présentons, dans ce qui suit, quatre espaces d'utilisation.

Pour accéder à l'application, l'utilisateur doit tout d'abord s'authentifier, pour cela il introduit son login et son mot de passe comme l'indique la figure [18]. Selon son profil ce dernier est redirigé vers son propre espace.

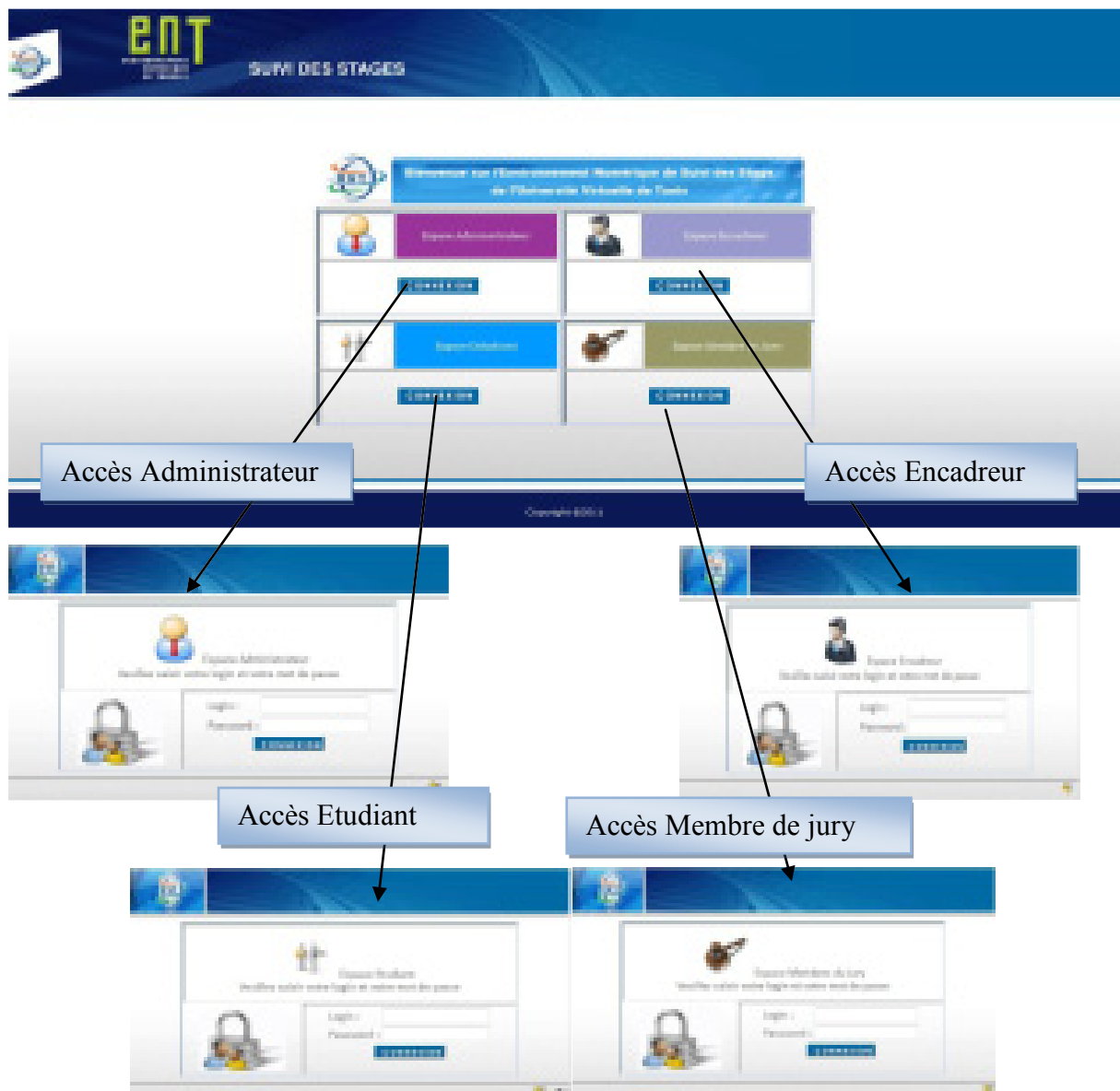


Figure 18: Authentification

## 1. Espace Administrateur « BackOffice »

### a. Page d'accueil

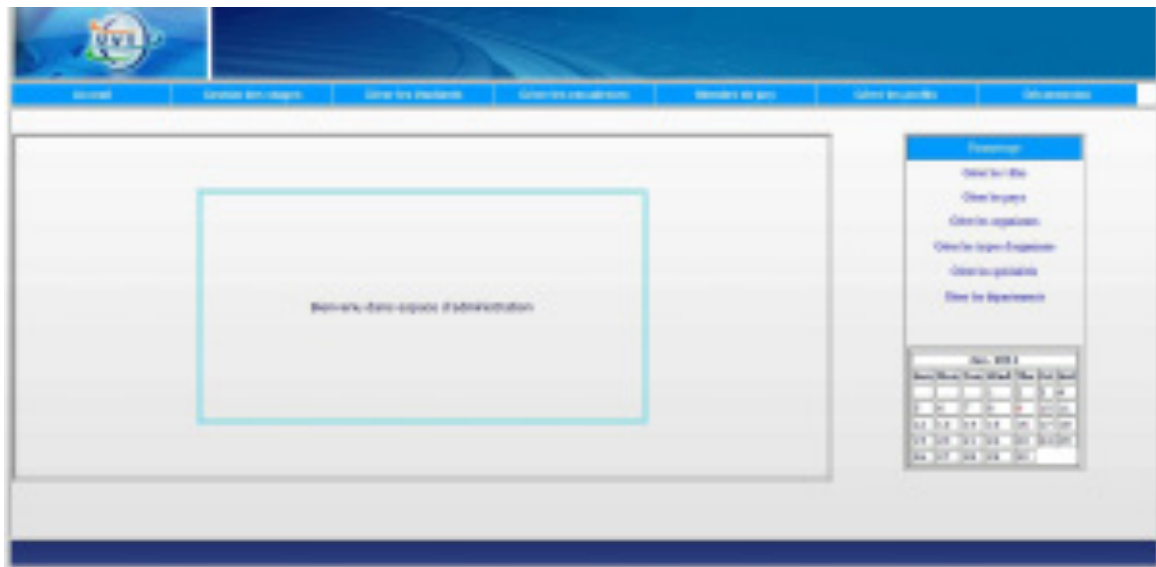


Figure 19 : Page d'accueil BackOffice

La page d'accueil de l'administrateur se présente comme l'indique la figure [19], à travers laquelle l'administrateur peut choisir une des options figurant dans le menu tel que l'ajout des stage, la gestion des étudiant, la gestion des encadreur, la gestion des profil...

### b. Interface « Ajouter un stage »

Parmi les fonctionnalités de notre application, l'Administrateur ou le gestionnaire peut créer un nouveau stage. Pour le faire, l'utilisateur doit remplir toutes les informations générales d'un stage comme indique la figure [20].

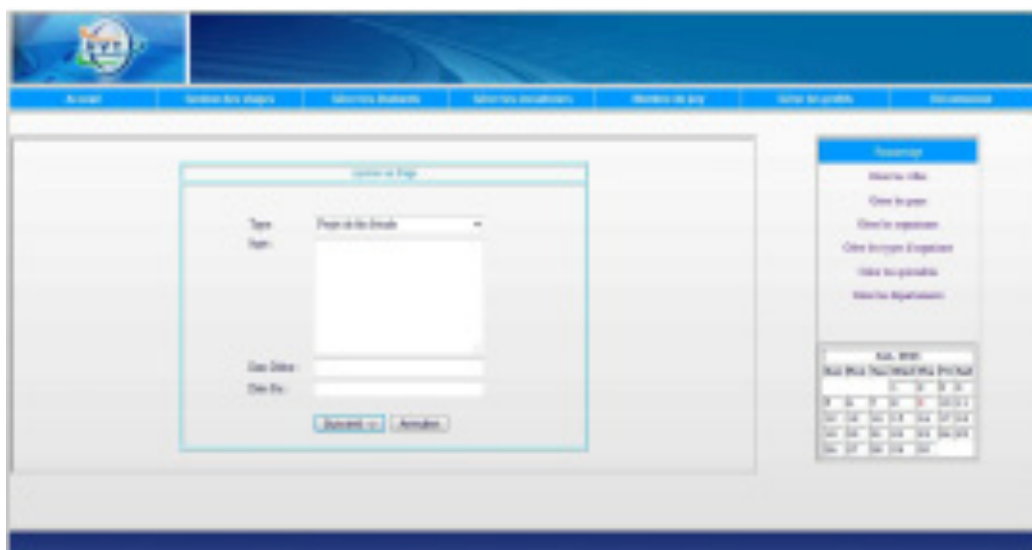


Figure 20 : Interface de l'Ajout de stage

Après avoir introduit les informations nécessaires d'un stage l'utilisateur est redirigé vers l'étape d'affectation des étudiants, où il peut affecter plusieurs étudiants pour le stage créé comme indique la figure [21].

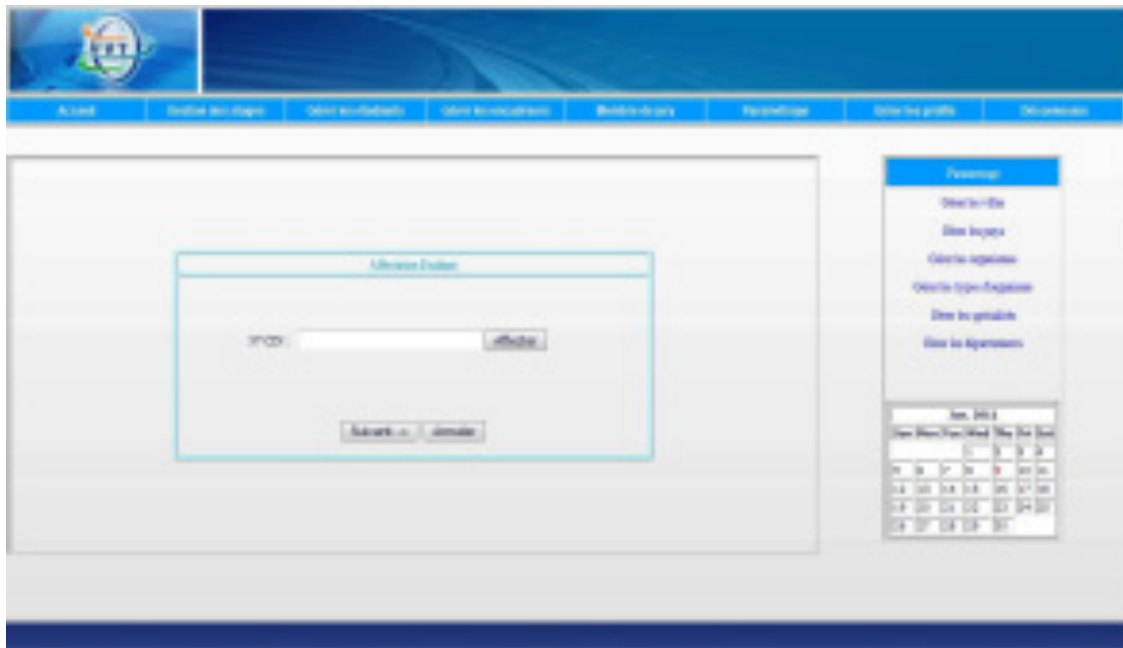


Figure 21: Interface d'affectation des étudiants

Par la suite, il sélectionne les encadreurs afin de les affecter à ce stage comme représente la figure [22].

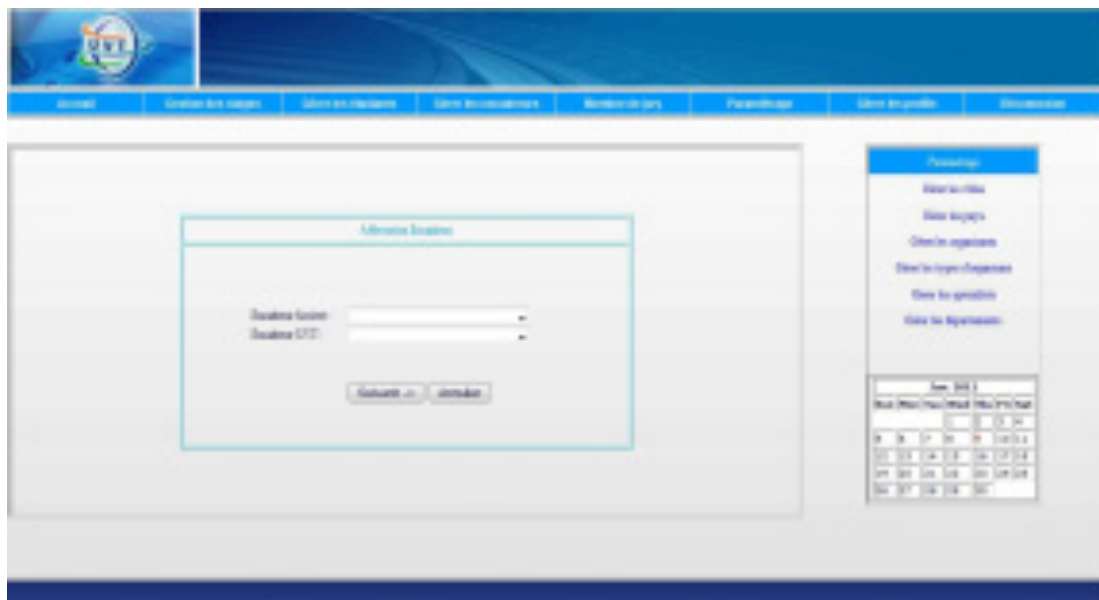


Figure 22: Interface d'affectation des encadreurs

En complétant les étapes indiquées précédemment, une fiche de stage contient toutes les informations du stage créé et sera affichée à l'utilisateur.

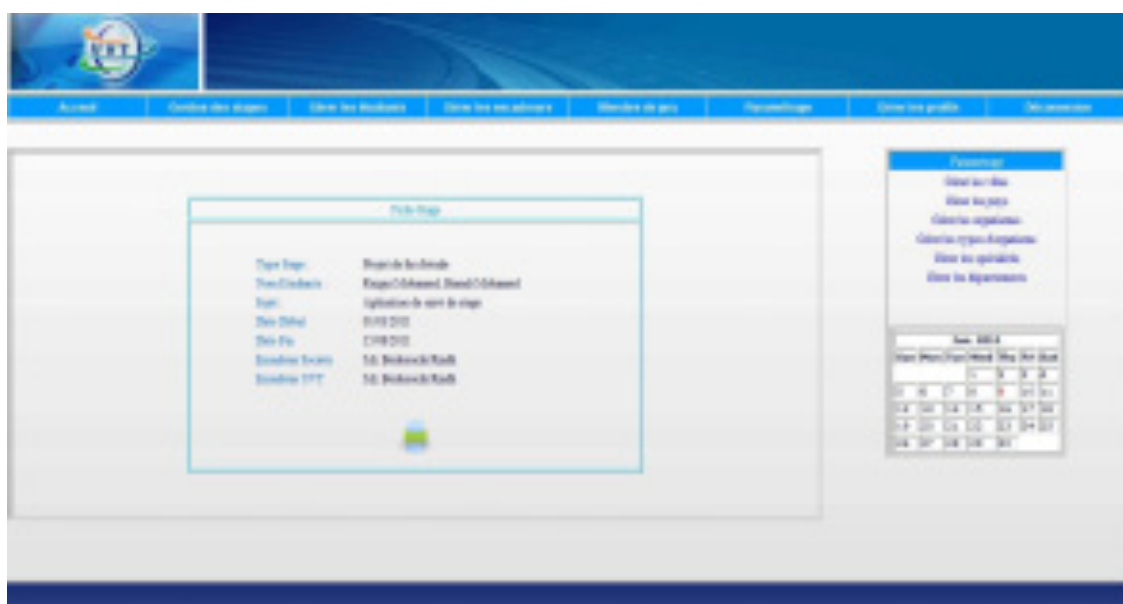


Figure 23: Fiche Stage

## 2. Espace Etudiant

### a. Page d'accueil



Figure 24: Page d'accueil de l'étudiant

La page d'accueil de l'étudiant se présente comme l'indique la figure [24], à travers laquelle plusieurs fonctionnalités seront mises à disposition de l'étudiant tel que le dépôt de document, le forum de discussion, la saisie des remarques...

### *b. Interface « Déposer un document »*

L'étudiant, a travers son espace, peut déposer plusieurs documents afin qu'ils seront téléchargeable par d'autre utilisateur, Pour le faire l'étudiant choisit a partir de son menus l'option déposer un document puis il parcourt le document à déposer comme indique la figure ci-dessous.

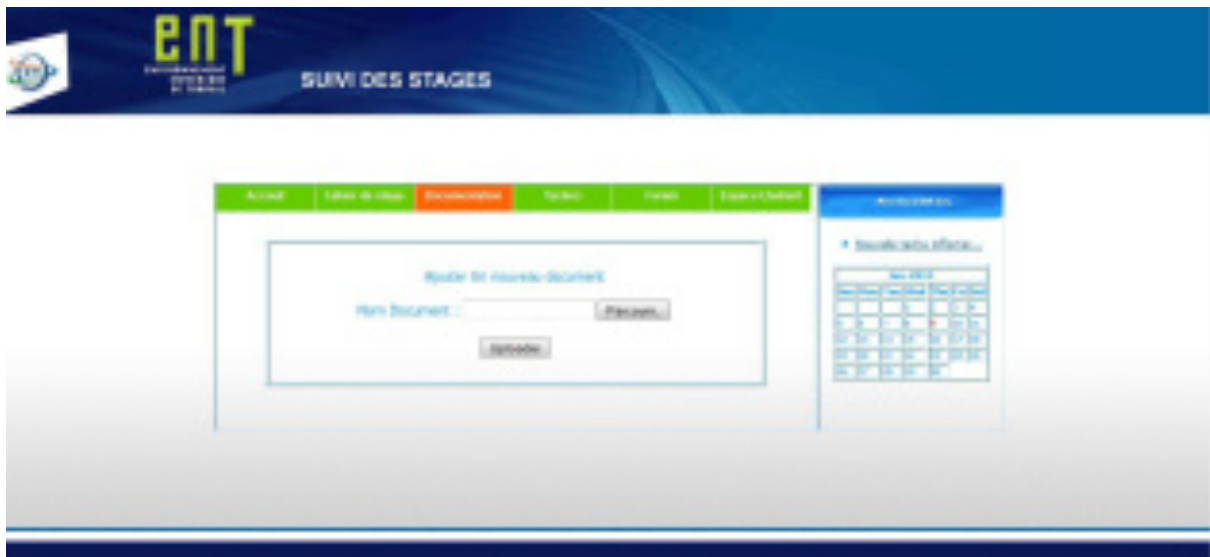


Figure 25: Interface de Dépôt d'un document

L'étudiant peut modifier ou supprimer le document déposé comme représenté dans la figure ci-dessous.

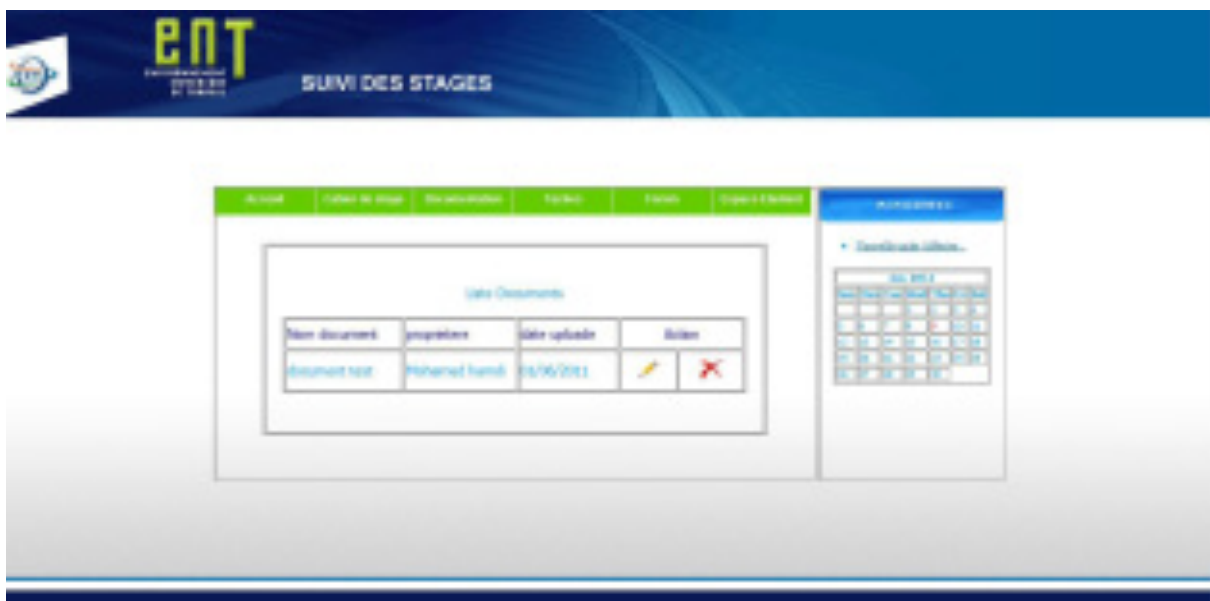


Figure 26: Interface de MAJ d'un document

### 3. Espace Encadreur

#### a. Page d'accueil

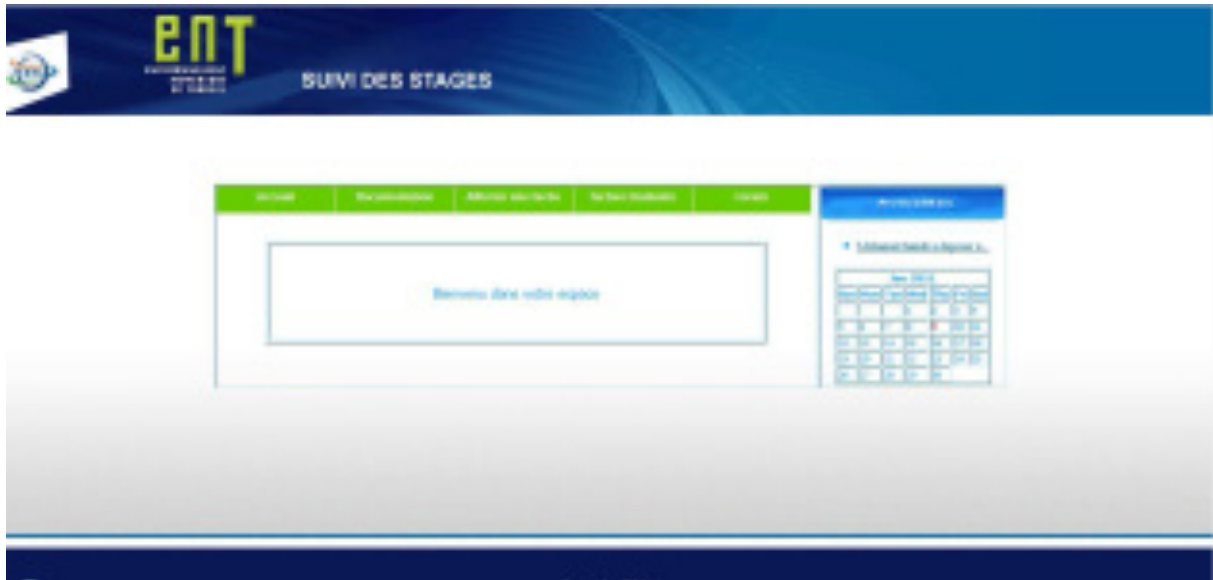


Figure 27: Page d'accueil de l'encadreur

Après s'être authentifié, l'encadreur va être redirigé vers la page d'accueil de son espace ou il peut choisir l'un des fonctionnalités figurant dans le menu général.

#### b. Interface « Affecter une tâche »

A partir de l'espace Encadreur, chaque encadreur peut affecter plusieurs tâches aux étudiants dans le cadre d'un stage. Pour le faire l'encadreur doit saisir la tâche à effectuer et choisir la date limite du dépôt de travail comme indique la figure ci-dessous.



Figure 28: Interface d'affectation des tâches

## 4. Espace Membre du Jury

### a. Page d'accueil



Figure 29: Page d'accueil du membre de jury

L'espace Membre du jury offre plusieurs fonctionnalités dans le cadre de suivi et de l'évaluation d'un stage tels que la validation des stages, la consultation des documents uploadé au cours de la période de stage...

Pour accéder à ces fonctionnalités le membre du jury doit choisir les options de menu générale dans la page d'accueil figurant ci-dessus.

### b. Interface « Valider un stage »

Les figures [30] et [31] représentent le processus de validation d'un stage.

Le membre du jury suit les actions suivantes afin de valider un stage :

- La sélection d'un stage ;
- Le choix d'un type de validation.



Figure 30: Interface de sélection du stage



Figure 31: Interface de validation du stage

## Conclusion

Dans ce dernier chapitre, il s'agit d'affiner ce que nous avons vu tout au long du rapport et simplifier le travail en des interfaces qui résument la future application de suivi des stages.



## Conclusion et perspectives

Notre projet a été réalisé dans le cadre d'un projet de fin d'études et qui a pour objectif le Suivi des stages au sein de l'Université Virtuelle de Tunis.

Notre travail se résume en la conception et la réalisation d'une application web permettant la gestion et le suivi des stages. Dans ce contexte, nous avons cherché à développer une application flexible et évolutive permettant son amélioration par la suite afin d'anticiper les changements continus des besoins des utilisateurs.

Cette application a permis, en premier lieu, d'améliorer la communication et l'échange de l'information et de la documentation entre l'étudiant, l'organisme d'accueil et l'université. Ainsi l'application permettra l'évaluation, le contrôle et le suivi des stages par plusieurs intervenants.

Pour la conception de notre application, nous avons eu recours à la méthodologie RUP. Cette approche nous a permis de bien comprendre la problématique et de bien modéliser les objectifs à atteindre. Ce qui nous a donné la possibilité de réaliser un système stable et évolutif.

Le projet s'est déroulé selon trois axes principaux afin de passer par les étapes essentielles de tout projet : l'analyse, la conception et la réalisation. Pour la réalisation, nous avons utilisé JAVA comme langage de programmation et MySQL comme système de gestion de base de données.

En outre, ce projet était une opportunité pour bien maîtriser le développement web et apprendre le langage JAVA ainsi que les outils et Framework liés tel que HIBERNATE, STRUTS et Myeclipse.

En guise de perspective Ce travail reste prêt pour toute amélioration envisageable comme réaliser le web service dédié à toutes les fonctionnalités de notre application pour qu'elle soit exploitable par des autres applications (exemple : Moodle).

## **Bibliographie et Nétographie**

### **Bibliographie**

[1] : Tutoriel JAAS Sur MyEclipse 6 17 AVRIL 2010

*Auteur : Riadh Bouhouch*

[2] : UML 2 en action, de l'analyse des besoins à la conception.

*Edition Eyrolles, février 2007, n° ISBN : 978-2-212-12104-9*

*Auteurs : Pascal Roques et Frank Vallée*

[3] : Java Persistence et Hibernate.

*Edition Eyrolles, février 2008, n° ISBN : 978-2-212-12259-6*

*Auteur : Antony patricio*

### **Nétographie**

[4] : Java pour le développement d'applications Web : Java EE

<ftp://ftp-developpez.com/mbaron/javaee/struts.pdf>

*Auteur : Mickaël BARON*

[5] : Struts par la pratique.

<http://tahe.ftp-developpez.com/fichiers-archive/struts.pdf>

*Auteur : Serge Tahe, université d'Angers.*

[6] : Premier pas avec Hibernate.

<http://www.scribd.com/doc/3165365/firsthibernateexampletutorialfr>

[7] : MyEclipse Hibernate Introduction Tutorial

<http://www.myeclipseide.com/documentation/quickstarts/hibernateintroduction/>