Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Gestion d'emploi du temps des soutenances

Réalisé par :

- **BENSAHLA -TANI Hidayet**
- **DIB Nesrine**

Présenté le 09 Juin 2014 devant la commission d'examination composée de MM.

- *Chouiti* *(Examinateur)*
- *ILES N* *(Examinateur)*
- *SMAHI Ismail* *(Encadreur)*

Année universitaire: 2013-2014

Dédicaces

Nous voulons présenter dans un premier temps nos dédicaces à notre encadreur « MR SMAHI Ismail ».

Nous voulons également lui témoigner notre gratitude pour sa patience et son soutien qui nous ont été précieux afin de mener notre travail à bon port, de tous nos respects aussi 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 soutenues tout au long de ce projet.

DIB Nesrine
&
BENSAHLA-TANI Hidayet

Remerciements

Nous adressons tout d'abord nos remerciements à Mr SMAHI Ismail 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 sa sagesse et ses précieux conseils, ce qui nous ont donné la force et le courage d'accomplir ce projet, il a encouragé également nos initiatives au travers de la grande liberté qu'il nous a autorisée.

Enfin un grand merci à nos chers parents, nos frères et sœurs pour leur bon humeur, sacrifices et surtout les aides morales et physiques que nous ont soumi pendant toute la période de la préparation de notre projet de fin d'études.

En fin, nous ne pouvons achever ce projet sans exprimer notre gratitude à tous les enseignants de mathématiques et informatique sans exception qui ont donné toutes leurs forces de nous apprécier les études.

ملخص

يسمح مشروعنا دراسة ،تصميم ،تحليل ،تنفيذ و تطبيق إدارة الوقت في تقديم المذكرات من التخصصات المختلفة. هذا الهدف يعمل على طريقة الاتصال وتبادل الوثائق والمعلومات بين أفراد المشروع، إنه يراقب أيضا أعمال الطلاب المقدمة من طرف الأساتذة المدربين، وأخيرا يتم تقديم المشاريع و تأييدها بعد إعطاء الملاحظات والتعليقات المقدمة من لجنة التحكيم في التاريخ المعين والغرفة المختارة.

لتحقيق هذا الهدف، قمنا بإنشاء تطبيق ويب في نتبينز في فئة الإنترنت جافا القائمة على تقنية ثلاثي الطبقات حيث يعتبر أوراكل ثين كقاعدة إدارة نظام البيانات ،و أ م ل لدراسة مفاهيمي.

ينبغي الإشارة إلى اداة باوردزانر التي ستساعدنا بشكل كبير في رسم وإدارة مختلف المخططات ام ل

الكلمات المفتاحية:

Netbeans, CSS, OracleThin, IDE, Apach Tomcat, JDK, JDBC, JPA, JSF, JSP,XHTML

Résumé

Notre projet permet l'étude, la conception, l'analyse et la réalisation d'une application de la gestion d'emploi du temps des soutenances de différentes spécialités. Cela à pour objectif la communication unidirectionnelle et l'échange de la documentation et de l'information entre les différents acteurs, il consiste également au suivi des taches des étudiants affectées par les encadreurs et finalement passer les soutenances en présentant les mémoires faits afin de les valider après avoir donné des remarques et commentaires par le jury dans une salle choisie et en date précise.

Pour atteindre cet objectif, nous avons créé une application web sous NetBeans dans la catégorie Java Web basé sur une architecture 3 tiers dont OracleThin comme système de gestion de bases de données, et une étude conceptuelle réalisée par le langage de modélisation UML.

Il faut noter que l'outil **Power designer** nous aidera énormément à dessiner et gérer les différents diagrammes UML.

Mots-clés : Netbeans, CSS, OracleThin, IDE, Apach Tomcat, JDK, JDBC, JPA, JSF, JSP, XHTML.

Abstract

Our project allows the study, design, analysis and implementation of an application of the employment time management defenses of different specialties. This target controlled way communication and exchange of documentation and information between objects, it is also monitoring spots students affected by the pro and finally pass defenses presenting submissions made to validate after giving remarks and comments by the jury in a selected specific date and room.

To achieve this goal, we have created a web application in NetBeans in the Java Web category based on a 3-tier architecture which OracleThin as management system database, a conceptual study by the UML.

Note thatthePowerDesignertool will helpus a lotto draw andmanage the variousUML diagrams.

Keywords: Netbeans, CSS, OracleThin, IDE, Apach Tomcat, JDK, JDBC, JPA, JSF, JSP, XHTML.

Table des matières

Introduction générale

La gestion est le pilotage de l'action collective au sein d'une organisation. Elle s'appuie notamment sur l'étude des organisations, objet des sciences de gestion.

La gestion désigne aussi l'ensemble du personnel responsable d'une entreprise ou d'une organisation .Quand nous parlons de la gestion nous trouvons un monde immense de domaines dont elle est utilisée.

Dans ce contexte, nous avons proposé de lancer un projet qui consiste à développer une application pour automatiser le processus des soutenances, en accompagnant tous les acteurs impliqués dans ce processus, en allant de la phase de proposition de sujet jusqu'à le jour de la soutenance.

L'objectif de ce projet est d'analyser, concevoir et développer une application de gestion des soutenances. Ce projet 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 desmémoires. En deuxième lieu, l'application doit assurer une partie de paramétrage qui intégrera la gestion des soutenances, la gestion des encadreurs, la gestion des étudiants, la gestion des profils de chaque utilisateur.

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

Dans le premier chapitre nous présentons les objets de persistance pour la création de la base de données que nous avons utilisé sous Netbeans ainsi nous décrivons les différentes architectures 3-Tiers.

Dans le chapitre qui suit nous abordons les deux phases « conception et analyse », au premier lieu nous mettons le cadre général se notre sujet. En second lieu, nous dégageons les contraintes que nous avons utilisées en réalisant notre mémoire ainsi quelques études préliminaires et les différents diagrammes.

Finalement, dans le dernier chapitre, nous expliquons les différentes étapes suivies pour faire l'application dans Netbeans en utilisant Java-Web et les différentes interfaces réalisées.

Chapitre 1 : Objet de persistance, pour la création de base de données

I. Introduction

Une des tâches les plus critiques pour les applications « Sauver et restaurer les informations ».

La persistance est la mémorisation des données depuis la mémoire vers des supports de stockage afin qu'elles soient restaurées quand l'application EST de nouveau exécutée. [1]

II. Abstraction

Presque toutes les applications ont besoin de données persistantes dans le temps. La persistance est un des concepts majeurs du génie logiciel. Sans persistance, après l'arrêt des machines, les systèmes sont souvent inexploitables.

III. Java et persistance des objets métiers

- La programmation orientée objets est aujourd'hui un standard. La persistance des objets est un point critique, dont dépendent la performance et la disponibilité des informations de tout projet Java. C'est pourquoi il est important de séparer les responsabilités d'accès aux données sérialisées des interfaces homme-machine.
  La couche métier ne doit contenir que la logique métier, c'est à dire propre à l'objet qu'elle représente.
  La couche DAO (Data Access Objects) a pour but de transformer les objets métiers en données sérialisées et inversement.
  La couche UI (User Interface) regroupe tout ce qui a trait à la présentation des données et aux interactions avec l'utilisateur, en appelant des traitements mis à disposition sous forme de méthode par les objets métier.

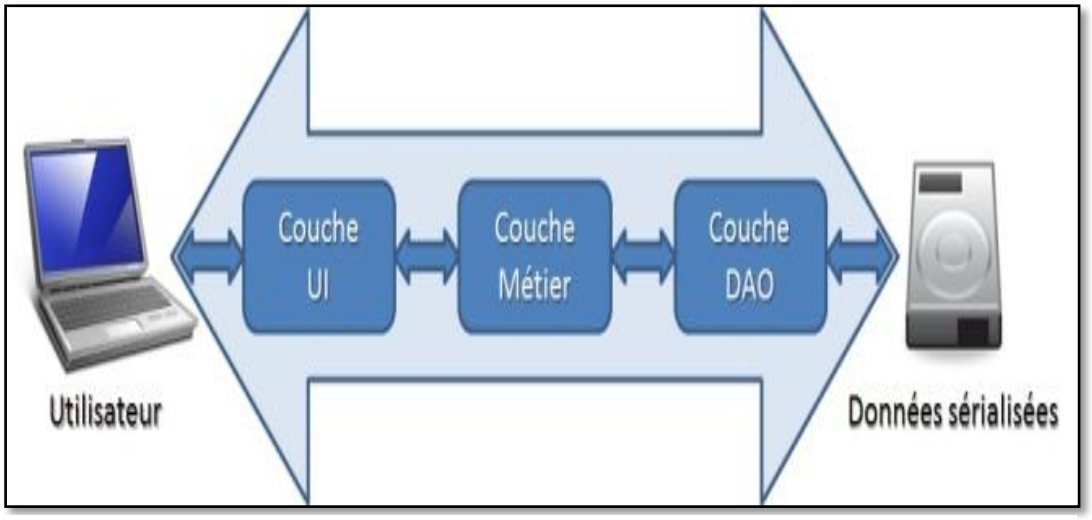


Figure 1: Architecture 3-Tiers[2]

- La sérialisation peut être assurée sur différents types de support (mémoire, fichiers, bases de données…)

Il est possible de développer ses propres outils avec les classes de sérialisation du JDK, mais c'est couteux à mettre en œuvre et à faire évoluer avec les versions du JDK.

De plus ces classes ne sont exécutables qu'en local. L'utilisation de bases de données permet de répondre à ces problématiques. Bien que les bases de données orientées objet telles Object Store[1] et DB4O[2] et les bases de données NoSQL soient l'outil de sérialisation idéal, ce sont les moteurs de bases de données relationnelles qui dominent le marché. Elles n'ont pas réussi à s'imposer face aux géants des bases de données relationnelles notamment à cause de la reprise de bases de données relationnelles déjà présentes dans les systèmes d'information des entreprises. Les développements basés sur l'utilisation directe des interfaces JDBC assurent un pont de bas niveau pour communiquer avec une base de données relationnelle en SQL.

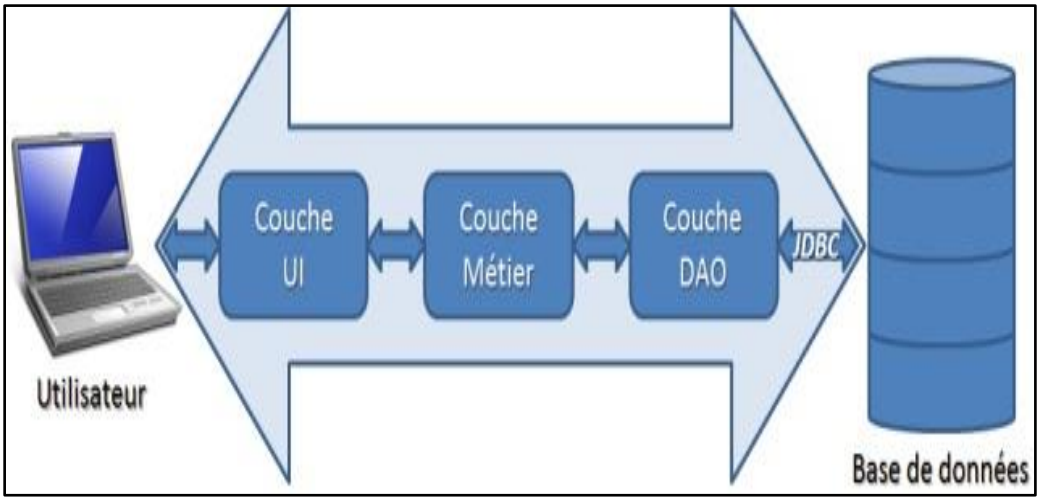


Figure 2: Architecture 3-Tiers avec JDBC[3]

- Dans ce cas, la couche DAO va faire correspondre de manière bijective

  - une table (appelée aussi relation) à une liste d'objets.
  - une ligne d'une table (appelée aussi tuple) à un objet.
  - un champ de base de données à un attribut d'objet.

[1] http://web.progress.com/fr/objectstore/

[2] http://www.db4o.com/

- une valeur d'un champ à une valeur d'attribut d'un objet Des modifications du modèle de données a des répercussions sur les requêtes aux bases, ce qui implique de faire évoluer individuellement les appels **jdbc**. La maintenance est de ce fait couteuse.

- Pour pallier à cette problématique et faciliter l'écriture de la couche DAO, il faut s'affranchir de la forme brute des données. La communauté Java a donc fait naître des frameworks de DAO, tels que Hibernate, Toplink, EclipseLink, Oracle BC4J... Ces ORM (Object Relational Mapping) permettent de se libérer d'une partie de la gestion des accès aux données. Le développeur de la couche DAO ne voit plus la couche JDBC ni les tables de la base de données. Il ne voit que l'image objet de la base de données fournie par la couche ORM.

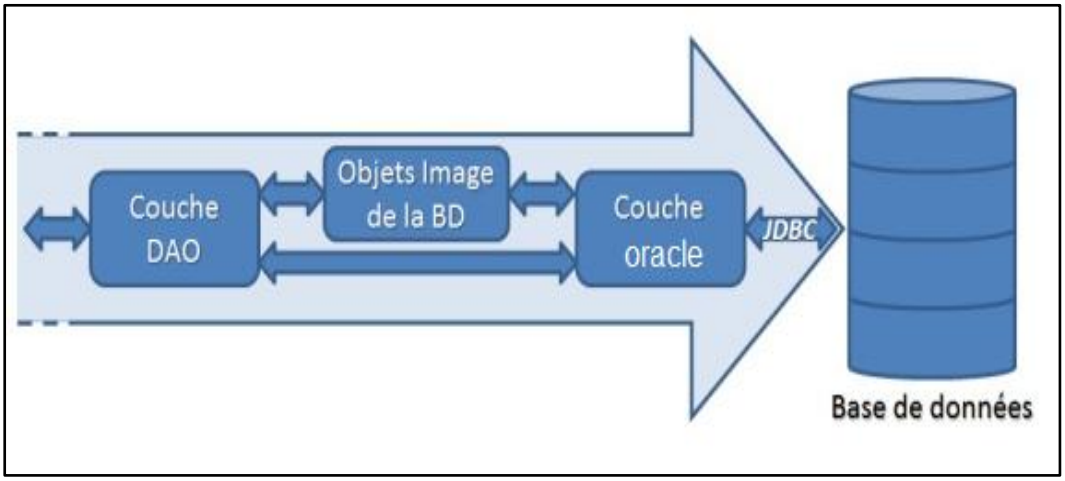


Figure 3: Couche DAO avec l'ORM Oracle[4]

- Comme ils ne reposent pas sur des standards Java, le choix de l'ORM n'est pas sans conséquences sur le code utilisé pour transférer une donnée métier dans la base de données. Il n'est en effet pas si aisé de passer d'Oracle à Hibernate par exemple.

Fort du succès des ORM, la firme Sun[3] décide de créer un standard pour l'utilisation de la couche ORM dans Java 5 : JPA (Java Persistence API). La couche DAO dialogue avec la spécification JPA qui est un ensemble d'interface du package **javax.persistence.** Quel que soit le produit qui implémente celle-ci, l'interface JPA utilisée dans la couche DAO reste la même. La couche DAO a ainsi été simplifiée et ne dépend plus du choix du fournisseur d'ORM ni du fournisseur de base de données puisqu'il est également possible d'utiliser certaines bases NoSQL telles que ObjectDB.

[3]SUN Micro System

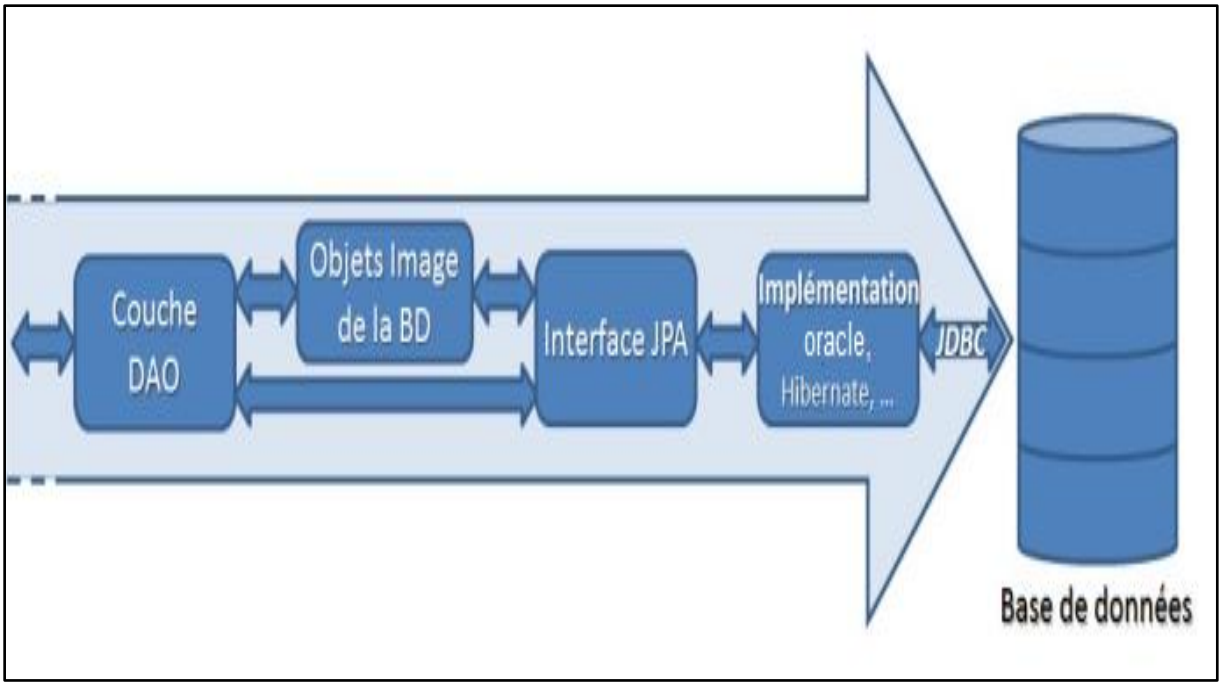


Figure 4: Couche DAO avec JPA [5]

- Bien avant JPA, Sun avait proposé JDO (Java Data Object), un standard pour assurer la persistance des objets Java dans tout type de système de gestion de ressources (bases de données relationnelles ou objets, fichiers XML, ...). Il existe même des implémentations pour des sources mainframe, JMS (Java Messaging Service) et web services.

En **comparant JDO et JPA** on peut s'apercevoir que JPA est un sous-ensemble de JDO. Et pourtant JDO n'a pas été un succès et finalement il existe plus d'implémentations JPA. Donc pour des raisons d'évolutivité et de maintenabilité, il vaut mieux privilégier l'utilisation de JPA. [6]

IV. Conclusion

Dans ce chapitre nous avons présenté les différents objets de persistance utilisés dans la réalisation de notre application, et le chapitre prochain nous allons définir la partie de conception et analyse en présentant le cadre fonctionnel et le cadre dynamique de ce système.

Chapitre 2 : Analyse et Conception

I. Introduction

Apres avoir présenté le cadre technique de notre application, à savoir les objets de persistance, nous essayons par le présent chapitre de détailler d'avantage notre démarche d'analyse et de conception.

Dans ce chapitre nous mettons le sujet dans son cadre général suivi d'une critique pour pouvoir dégager les contraintes à respecter pendant la réalisation denotreprojet, ainsi présenter l'analyse et la conception que nous avons adopté en respectant la démarche du processus unifié (UP).

II. Processus unifié

1. Processus de développement logiciel

Le succès spectaculaire d'UML ne doit pas faire oublier qu'il ne s'agit que d'un langage de modélisation graphique, dont la vocation n'est pas de couvrir tous les aspects du génie logiciel. Complément idéal d'UML, un processus de développement logiciel tel que le processus unifié a précisément pour but de spécifier les différentes phases d'un projet, de l'élaboration du cahier des charges au déploiement de l'application. [7]

2. Processus unifié

a. Définition

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.[8]

b. Objectifs

Le processus unifié fournit un cadre au développement logiciel pour la construction de systèmes orientés objet.

Le processus unifié répond aux exigences fondamentales suivantes :

- être guidée par les besoins des utilisateurs
- être centrée sur l'architecture logicielle

- être itérative et incrémentale

a. ***Les phases***

Les phases d'un processus de développement sont des états de celui-
ci à uninstant « t ».Le cycle de développement du Processus Unifié organise les tâches et les it érations en quatre phases :

- **Analysedes besoins** : spécification des besoins et aussi une sorte d'étude faisabilité ou on effectue les recherches nécessaires pour décider si on poursuit ou non le projet
- **Élaboration** : on développe de façon incrémentale l'architecture du noyau, les risqueset la plupart des besoins sont identifiés
- **Construction** : on construit des sous ensembles exécutables et stables du produit final
- **Transition** : le produit final est livré en version beta à la disposition des utilisateurs

c. Les activités

Les activités représentent les actions à effectuer au cours d'une phase ; une phase passe par l'ensemble des activités.

Ses activités de développement sont définies par 6 disciplines fondamentalesqui décrivent la modélisation métier, la capture des besoins, l'analyse et laconception, l'implémentation, le test et le déploiement [9].

La répartition de l'effort change avec le temps. Les première itérations ont tendance à mettre l'accent sur certaines activités (analyse et conception), les autres itérations moins. Les efforts seront ensuite portés sur l'implémentation, les tests et le déploiement.

III. Etude préliminaire

Dans cette partie nous nous intéressons à présenter en détail notre démarche d'analyse, a savoir l'identification des acteurs, identification des messages, diagramme de cas d'utilisation et le diagramme de classes

1. **Identification des acteurs**

Les acteurs sont des entités externes qui interagissent avec le système, comme une personne humaine ou un robot. Une même personne (ou robot, ...) peut être plusieurs acteurs pour un système, c'est pourquoi les acteurs doivent surtout être décrits par leur rôle, ce rôle décrit les besoins et les capacités de l'acteur. Un acteur agit sur le système (accès de lecture/écriture).

L'activité du système a pour objectif de satisfaire les besoins de l'acteur. Les acteurs sont représentés par un pictogramme humanoïde sous-titré par le nom de l'acteur. [10]

Dans notre application, nous avons défini quatre acteurs qui sont:

L'étudiant

L'étudiant choisit un sujet parmi les sujets proposés, puis il réalise les taches affectées par son encadreur à l'aide des recherches et de la documentation durant la période de préparation de mémoire en respectant la date limite de la remise du travail demandé. Enfin il pose son projet et passe sa soutenance.

L'encadreur

Chaque encadreur propose des sujets pour la préparation des projets de fin d'études, puis il affecte plusieurs taches aux étudiants après avoir choisi leurs sujets et consulte leurs travaux. Le suivi du travail des étudiants sera effectué par l'ajout de remarques et de commentaires sur les mémoires et finalement il autorise les étudiants pour se soutenir.

Le jury

Le jury consulte les mémoires des étudiants et donne des remarques concernant leur travail et leur présentation en assistant à la soutenance, et enfin la valide.

L'administrateur

L'administrateur assure le paramétrage de notre application. Cette partie intégrera la gestion des soutenances, la gestion des encadreurs, la gestion des étudiants, la gestion des profils de chaque utilisateur, ainsi que la préparation des salles des soutenances et la classification des mémoires élaborés par les étudiants.

2. **Identification des messages**

Un message représente la spécification d'une communication unidirectionnelle entre objets et qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur. [11].Un message est normalement associé à deux occurrences d'évènement : un évènement d'envoi et un évènement de réception. Les messages répertoriés entre le système et ses acteurs sont :

Messages	**Réponses**
Demander un sujet	Envoyer le sujet
Ajouter un sujet	Valider l'ajout
Supprimer un sujet	Valider la suppression
Demander l'autorisation de soutenance	Valider l'autorisation de soutenance
Déposer le mémoire	Mémoire validé

3. **Les diagrammes de cas d'utilisation**

Le diagramme de cas d'utilisation permet de décrire l'interaction entre l'acteur (les acteurs) et le système. L'idée forte est de dire que l'utilisateur d'un système logiciel a un objectif quand il utilise le système.

Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système. Les use case (cas d'utilisation) sont représentés par une ellipse sous-titrée par le nom du cas d'utilisation (éventuellement le nom est placé dans l'ellipse). Un acteur et un cas d'utilisation sont mis en relation par une association représentée par une ligne.[12]

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

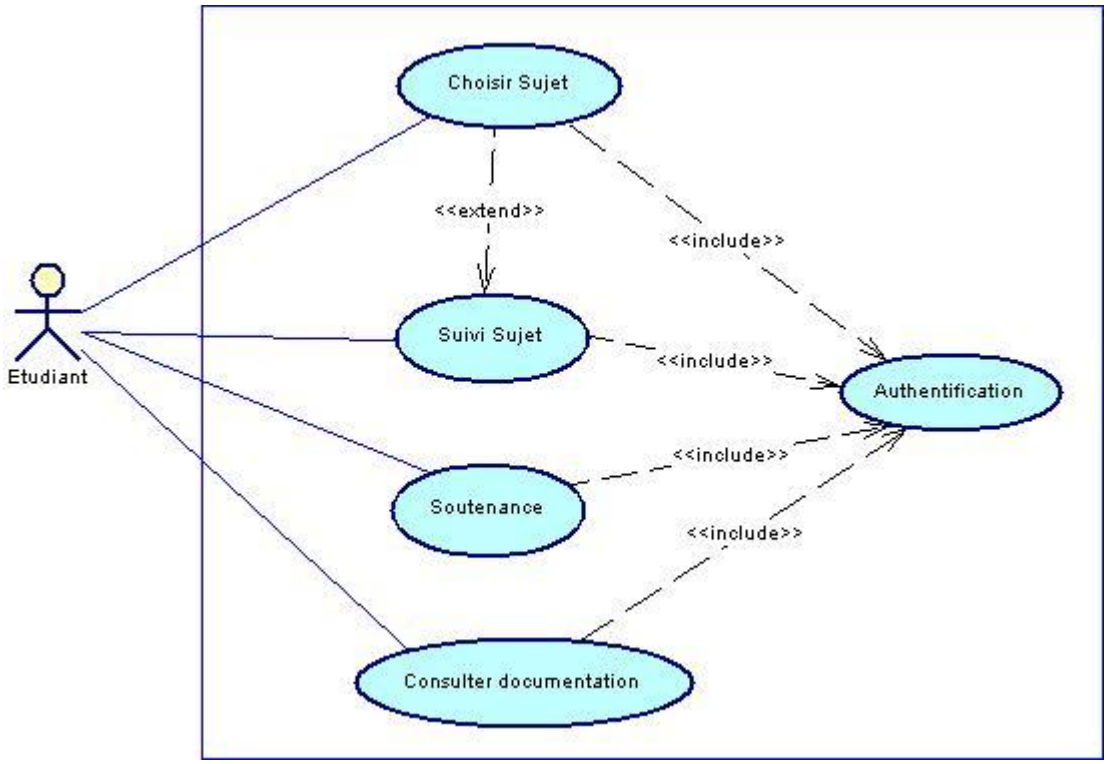


Figure 5 : Diagramme de cas d'utilisation de l'étudiant

- *Description ducasd'utilisation « choisir un sujet»*

SOMMAIRE D'IDENTIFICATION	
Titre	Choisir un sujet
Acteur	L'étudiant
But	Préparer le projet de fin d'étude
Résumé	L'étudiant choisit le sujet puis valide l'action
DESCRIPTION DES ENCHAINEMNT	
Pré condition	**Post condition**
L'étudiant doit être authentifié	le sujet est choisi
ENCHAINEMENT NOMINAL	
• L'étudiant choisit le sujet • L'étudiant valide l'action • Le système vérifie le type de choix • Le système uploade le choix sur le serveur • Le système met à jour la base de données	

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

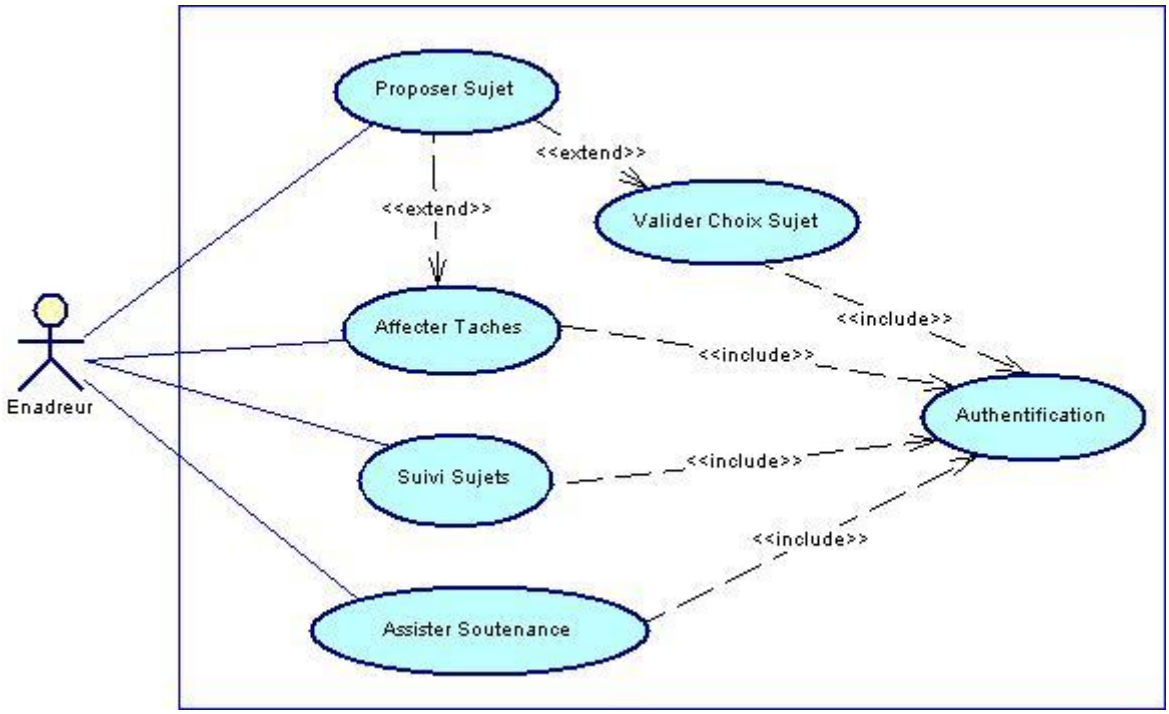


Figure 6 : diagramme de cas de l'encadreur

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

SOMMAIRE D'IDENTIFICATION	
Titre	Affecter une tâche
Acteur	L'encadreur
But	Assister les étudiants pour bien réaliser leur projet de fin d'étude
Résumé	L'encadreur remplitle formulaired'affectation detâcheet validel'action. Le système enregistre la tâche dans la base de données.
DESCRIPTION DES ENCHAINEMNT	
Pré condition	**Post condition**
L'encadreur doit être authentifié	La tâche est affectée
ENCHAINEMENT NOMINAL	
• L'encadreur remplit le formulaire • L'encadreur valide l'affectation • Le système enregistre la tâche dans la base de données	

- *Diagramme de cas d'utilisation de jury*

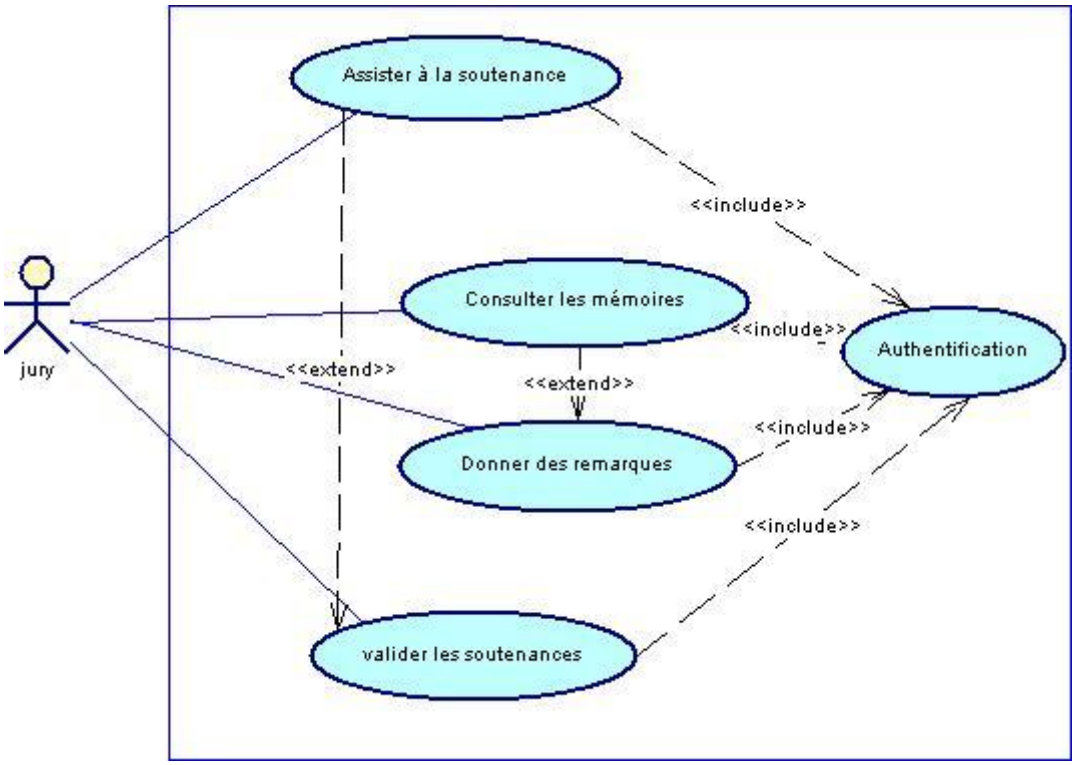


Figure 7 : Diagramme de cas d'utilisation de jury

- *Description du cas d'utilisation « Valider une soutenance»*

SOMMAIRE D'IDENTIFICATION	
Titre	Valider une soutenance
Acteur	Le jury
But	Valider les soutenances des étudiants
Résumé	Le jury remplit le formulaire de validation et validel'action.Le système enregistre la tâche dans la base de données
DESCRIPTION DES ENCHAINEMNT	
Pré condition	**Post condition**
Le Jury doit être authentifié	La soutenance est modifiée
ENCHAINEMENT NOMINAL	
• Le Jury remplit le formulaire • Le Jury valide l'action • Le Système enregistre la tâche dans la base des données	

- *Diagrammede cas d'utilisationde l'administrateur*

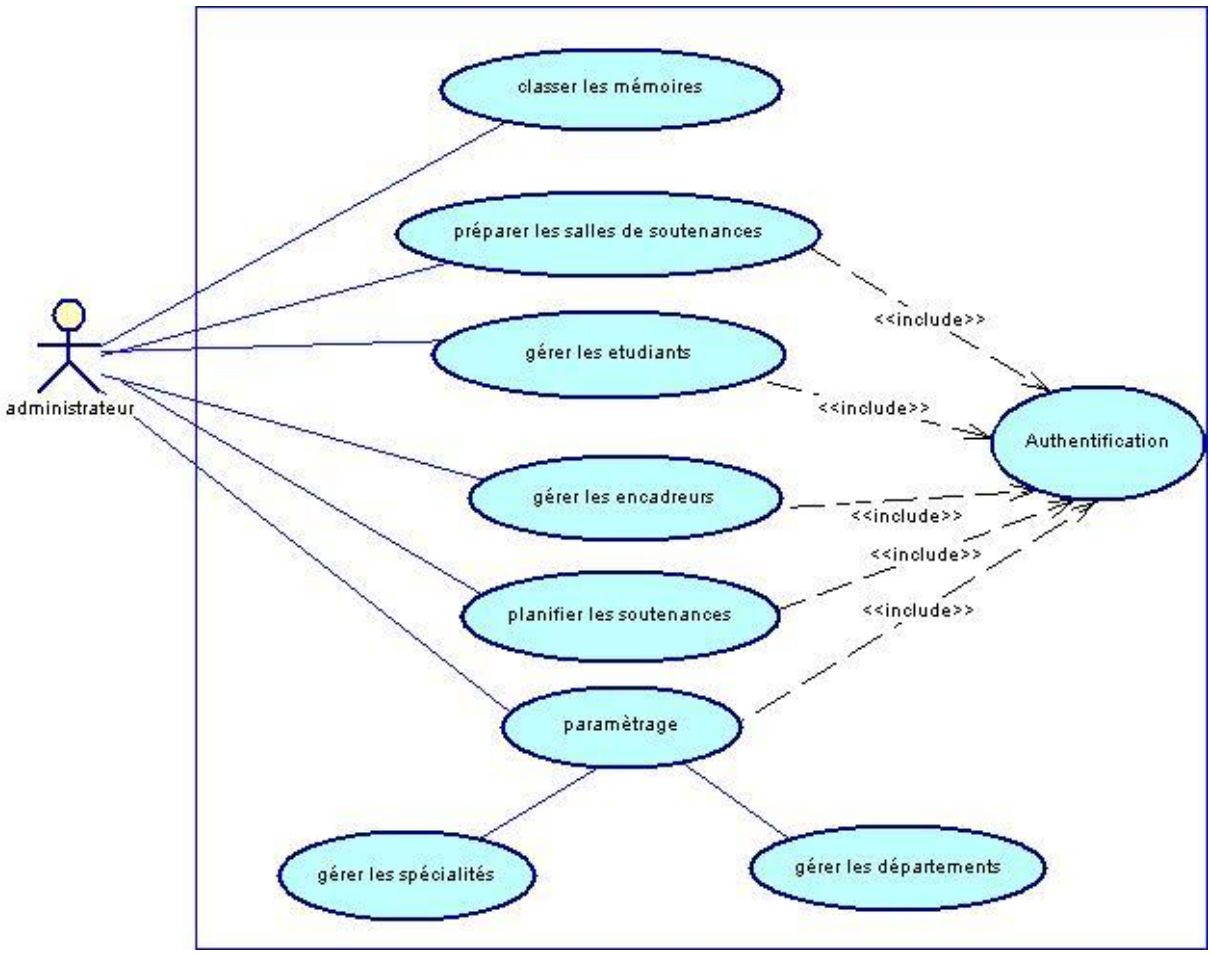


Figure 8 : Diagramme de cas d'utilisation de l'administrateur

- Description du cas d'utilisation « Ajouter une soutenance»

SOMMAIRE D'IDENTIFICATION	
Titre	Ajouter une soutenance
Acteur	L'administrateur
But	Ajouter les soutenances des étudiants
Résumé	L'administrateur remplit le formulaire d'ajout des soutenances et validel'action. Le système enregistre la tâche dans la base de données
DESCRIPTION DES ENCHAINEMNT	
Pré condition	**Post condition**
-L'administrateur doit être authentifié	- La soutenance est ajoutée
ENCHAINEMENT NOMINAL	
• L'administrateur remplit le formulaire • L'administrateur valide l'ajout	

- Le Système enregistre la tâche dans la base des données

4. **Diagramme de Classes**

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Chaque langage de Programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier. [13]

a. Identification 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 département** : contient les différents départements de l'université considérés.
- **Classe enseignant** : contient les enseignants de différentes spécialités.
- **Classe Est-constitué** : contient le jury et le nom d'enseignant avec sa qualité dans ce jury.
- **Classe Est-soutenu** : indique pour chaque jury le sujet à examiner ainsi la salle de soutenance, la date et l'heure.
- **Classe étudiant** : contient tous les étudiants qui passent leurs soutenances.
- **Classe jury** : contient toutes les informations sur le jury.
- **Classe salle-soutenance** : contient toutes les informations sur la salle de soutenance.
- **Classe spécialité** : contient les différentes spécialités de l'université.
- **Classe sujet** : contient les différents sujets affectés par les encadreurs et présentés pas les étudiants

b. Classes et attributs

Département (code_depart, nom_depart, tel_depart)
Enseignant (code_enseig, nom_enseig, prenom_enseig, #code_depart)
Est _constitue (code_enseig, code_jury, qualité)
Est _soutenu (code_sujet, code_jury, code_salle, heure_sout, date_sout)
Etudiant (code_etud, nom_etud, prenom_etud, #code_sujet)
Jury (code_jury, num_sequentiel)
Salle_de_soutenance (code_salle, num_salle)
Spécialité (code spécial, nom_spécial)
Sujet (code_sujet, nom_sujet, #code_enseig, #code_spécial)

c. Diagramme des classes

Ce diagramme est le diagramme de classes abordé dans le développement de notre application en présentant les différentes classes.

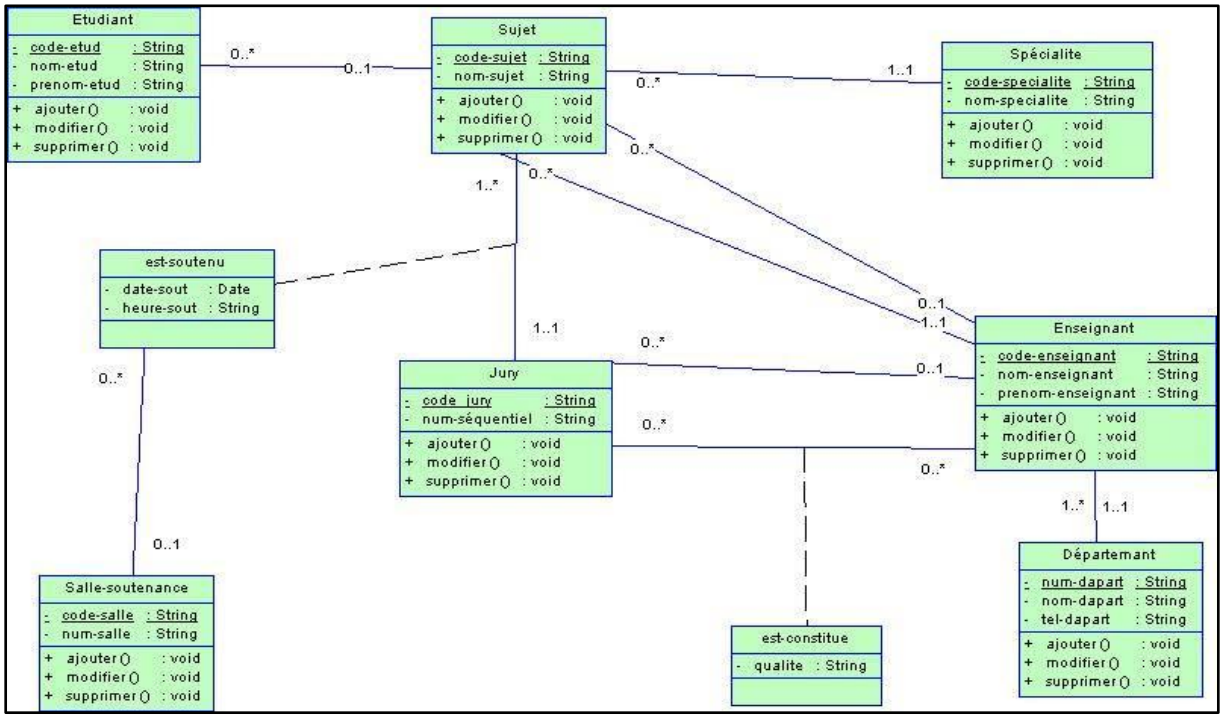


Figure 9 : Diagramme de classe

IV. Analyse

1. Modèle dynamique (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 Unified Modeling Language. [14]

- *Diagramme de choix de sujet*

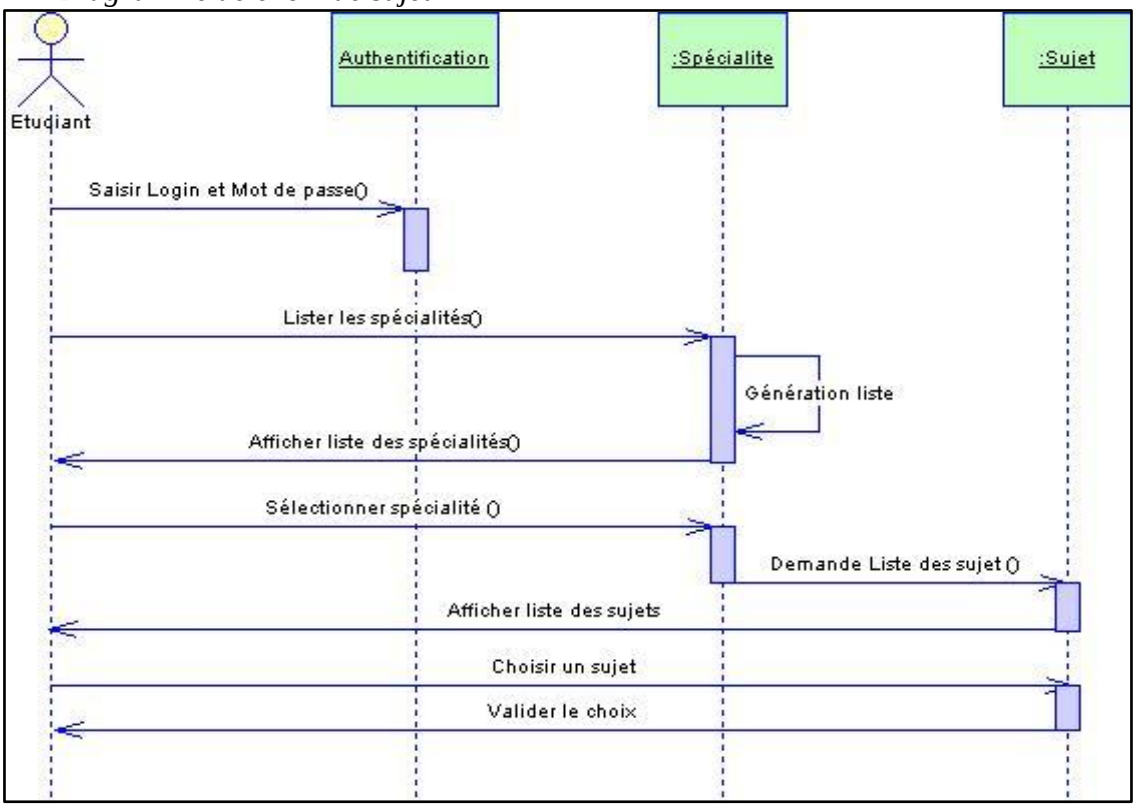


Figure 10 : Diagramme de séquence « choix sujet »

- *Diagramme « affecter un sujet »*

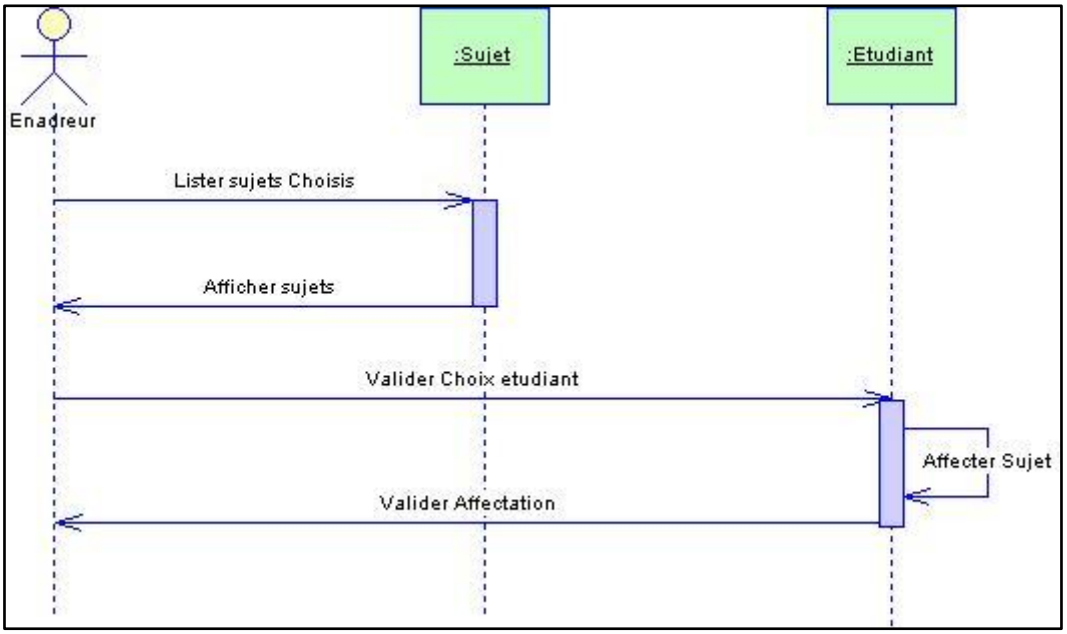


Figure 11 : Diagramme de séquence affecter un sujet

- *Diagramme « valider une soutenance »*

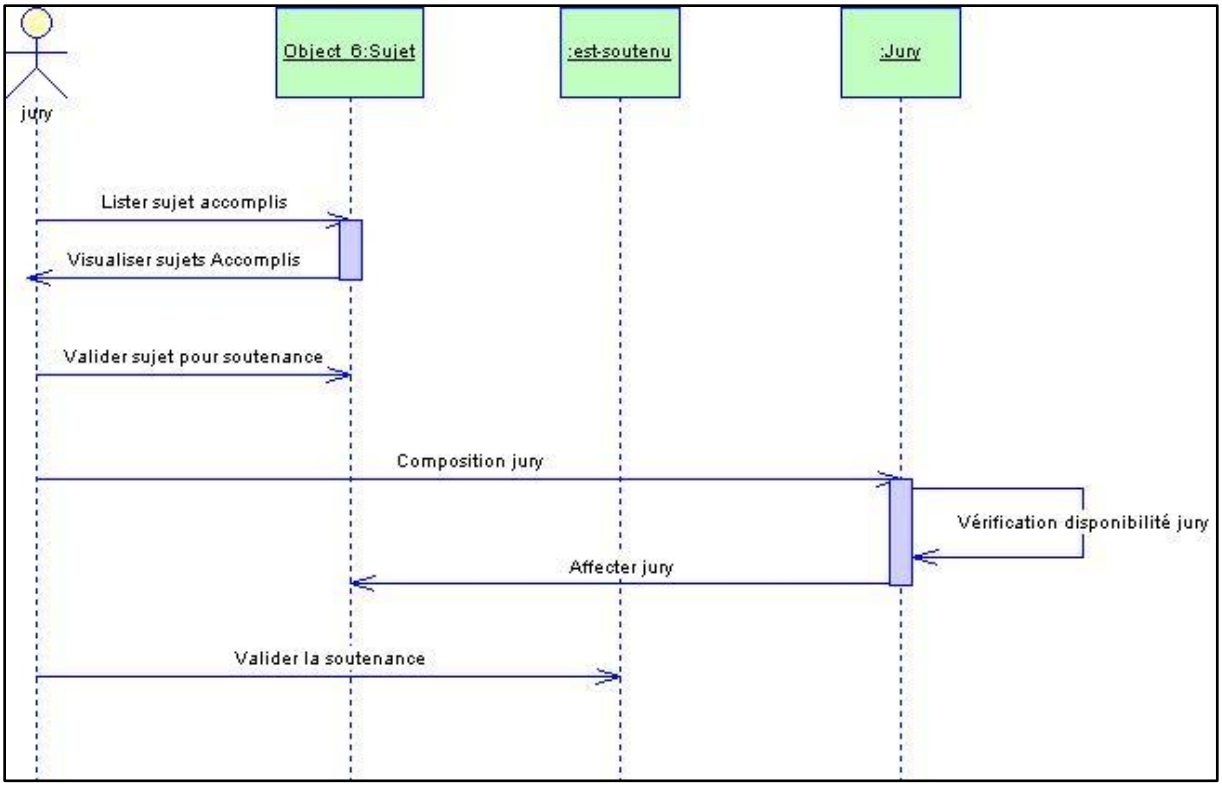


Figure 12 : Diagramme de séquence valider une soutenance

V. Passage de diagramme de classe vers le MCD

1. Model conceptuel de données(MCD)

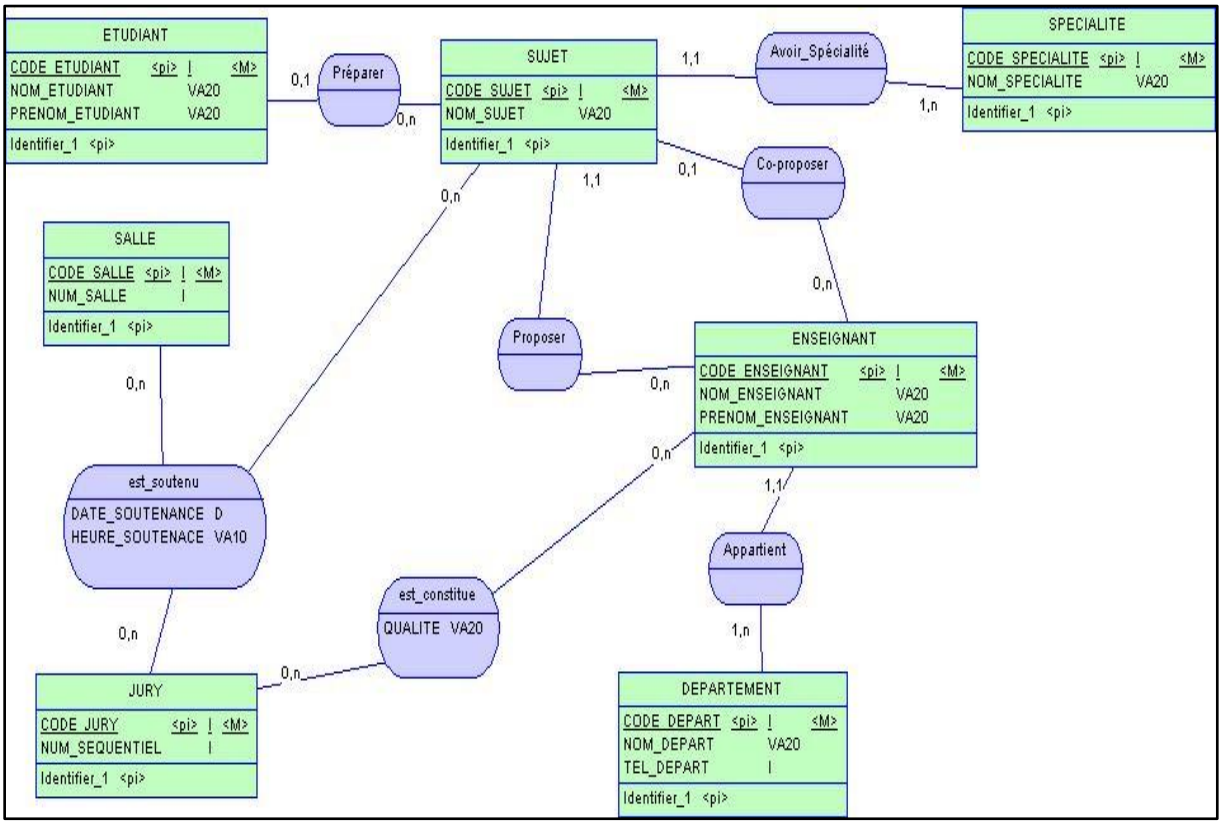


Figure 13:Model Conceptuel de Donnée

2. Passage de diagramme de classe vers le MCD

Pour pouvoir générer le Model Conceptuel de Données (MCD) à partir du Diagramme de Classes que nous avons établi, nous avons suivi un ensemble de règles qui permettent la création du MCD à partir du diagramme de classes. Ces règles sont représentées dans letableau qui suit:

UML	Entité-Association
Classe	Entité
Association (relation)	Association
Objet (instance)	Occurrence
Multiplicité	Cardinalité
Diagramme de classe	Modèle Conceptuel de données
0..1	0, 1
1..1 ou 1 (le défaut)	1, 1
0..* ou * (le défaut)	0, N
1..*	1, N
N, N	*..*

VI. Conclusion

Dans ce chapitre, nous avons présenté toute la conception et l'analyse de notre système. Il a été consacré à la modélisation de l'aspect statique et dynamique de notre application, en sebasant 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 : Implémentation

I. Introduction :

Après avoir entamé la partie de conception et défini les objets de persistances, ce chapitre présente les étapes nécessaires pour réaliser notre projet concernant une application de la gestion d'emploi du temps des soutenances dans les universités sous NetBeans.

II. Initialisation du projet NetBeans :

NetBeans est basé sur la notion de projet. Un projet correspond à tout ou partie d'une application que l'on souhaite développer. Il contient à la fois les sources, c'est-à-dire les fichiers contenant le code, et l'ensemble des informations de configuration de l'application.

Pour cette application, nous devons créer un projet de type Web Application dans NetBeans.

- Aller dans le menu Fichier > Nouveau Projet. Choisir Web Application dans la catégorie Java Web.
- Dans l'écran Name and Location, nommer le projet.
- Dans l'écran Server and settings, laisser les valeurs par défaut.
- Dans l'écran Framework, cocher Java Server URL Pattern. Puis, cliquer sur Terminer.

Cette dernière étape de configuration est nécessaire afin de pouvoir utiliser Java Server Faces (JSF) pour le développement des pages web de l'application.

III. Page de démarrage :

La toute première page que voit l'utilisateur lorsqu'il utilise une application web est la page d'accueil. Il est nécessaire de définir de quelle page il s'agit pour l'application « Gestion de Soutenances ».

Par défaut, lors de la création du projet, NetBeans créée une page index. Html dans le répertoire Web Pages et la définit comme page d'accueil dans le fichier web.xml, comme cela est montré par la figure ci-dessous.

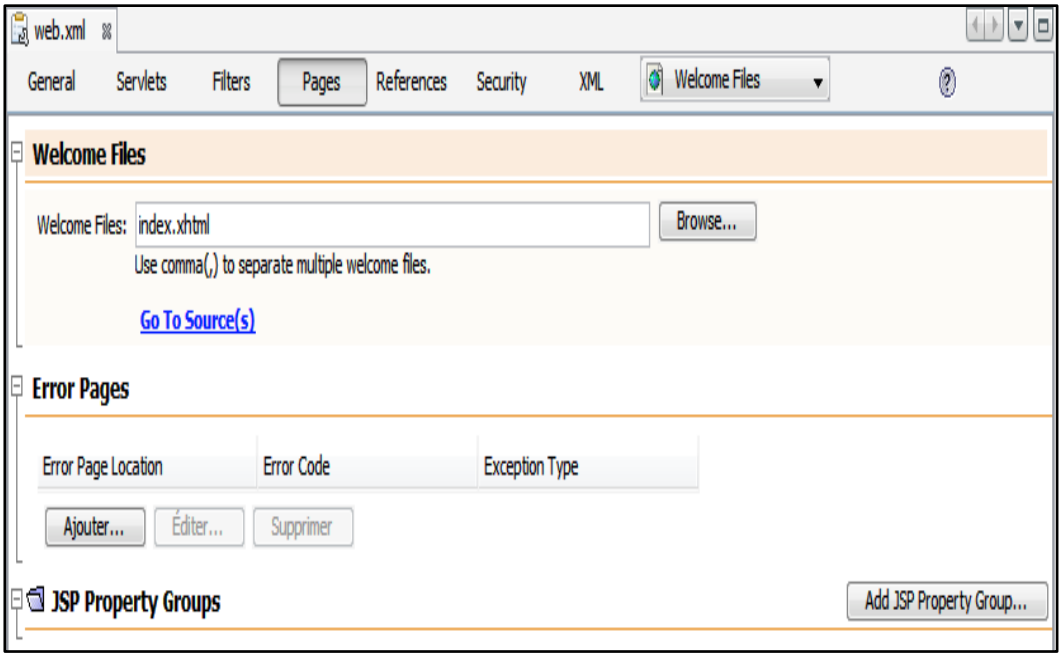


Figure 14:page de démarrage

IV. Prise en main du serveur de bases de données

1. Démarrage du serveur de bases de données Java DB

Dans l'onglet Services, cliquer sur Bases de Données.

Cliquer sur Java DB puis avec le clic-droit de la souris choisir Démarrer le serveur dans le menu contextuel (si le serveur est déjà démarré, cette option apparaîtra en grisé).

Vous avez ainsi accès au serveur Java DB qui tourne sur votre machine : en cliquant sur la croix (ou la flèche) qui se trouve à gauche de Java DB, vous accédez à une vue de l'ensemble des bases de données hébergées par le serveur.

Pour le moment la base de données de l'application « Gestion de soutenance » n'existe pas, nous allons la créer.

2. Création de la base de données « Gestion de Soutenance » et import de données

Aller à *Bases de Données ->Java DB* et choisir *Créer une base de données* dans le menu contextuel (clic-droit avec la souris).

Saisir "Gestion de soutenance" pour le nom de la base de données, choisir un nom d'utilisateur et un mot de passe.

Cette opération créée la base de données sur le serveur et créée une connexion entre NetBeans et cette base. Celle-ci apparait sous la forme **jdbc:oracle://localhost:1527/Gestion de soutenance** dans la liste des bases de données, un peu plus bas dans l'onglet *Services* > *Bases de données.*

Cliquer sur **jdbc:oracle://localhost:1527/Gestion de soutenance** et choisir Exécuter la commande... dans le menu contextuel.

Dans l'éditeur de commandes SQL qui s'est ouvert, coller le contenu du script automatique de création des tables et des données que nous avons réalisé pour vous. Puis cliquer sur l'icône Exécuter la commande SQL comme indiqué sur la figure ci-dessous.

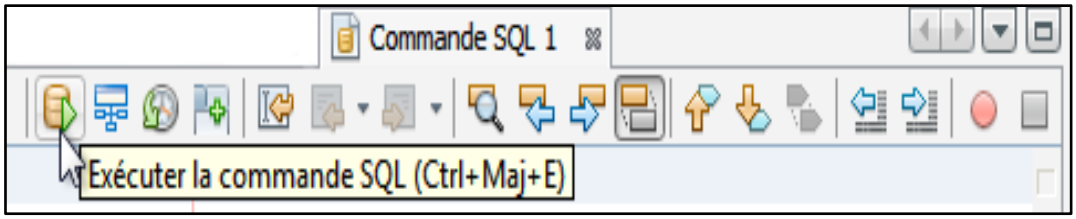


Figure 15 : Exécution SQL sous NetBeans

3. Affichage du contenu de la base de données « Gestion de Soutenance »

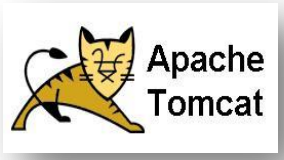
Cliquer sur la croix (ou sur la flèche) à gauche de « **Gestion de Soutenance** » dans **jdbc:oracle://localhost:1527/ Gestion de Soutenance** pour faire apparaître son contenu, puis faire apparaître de la même manière le contenu de **GestionSoutenanceDB** (le nom de la base de donnée) puis Tableaux (ou Tables).

Le répertoire Tableaux (ou Tables) contient les tables de la base. En développant les items vous pouvez voir les différentes colonnes de chaque table.

La base de données est maintenant configurée.

V. Configuration de l'accès à la base de données

1. Le serveur web Apach Tomcat



Apache Tomcat est un conteneur web léger, gratuit, libre de servlets et JSP Java EE.

C'est un projet principal de l'*Apache Software Foundation*. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiersXML et de propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP. Il a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant

2. Connecteur JDBC

Pour accéder à la base de données, JPA utilise un **connecteur**. Pour les bases de données relationnelles, le connecteur utilisé est JDBC. Il faut configurer ce connecteur de la manière suivante :

- Lui donner un nom unique permettant à l'application de le référencer comme une source de données.
- Définir un pool de connexion, c'est-à-dire un ensemble de liens prédéfinis entre le serveur d'applications et le serveur de bases de données qui pourront être utilisés par l'application. La configuration du pool de connexion consiste à donner l'adresse du serveur de bases de données, le nom de la base à laquelle accéder ainsi qu'**un login/mot de passe d'accès.**

3. Jdk 7

Java Développent Kit Java est l'environnement dans lequel le code Java est compilépour être transformé en byte code afin que la machine virtuelle JAVA (JVM) puissel'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 lebyte code Java).
- jar – l'archiveur, qui met sous forme d'un paquetage unique l'ensemble des fichiersclass en un fichier JAR ;
- javadoc– le générateur de documentation, qui génère automatiquement de ladocumentation à partir des commentaires du code source ;
- jdb – le débogueur.

4. System de gestion de base de donnée Oracle

Oracle Database est un système de gestion de base de donnéesrelationnel (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO). Fourni par Oracle Corporation, il a été développé par Larry Ellison, accompagné d'autres personnes telles que Bob Miner et Ed Oates.

5. Créations de classes entités à partir d'une base de données existante

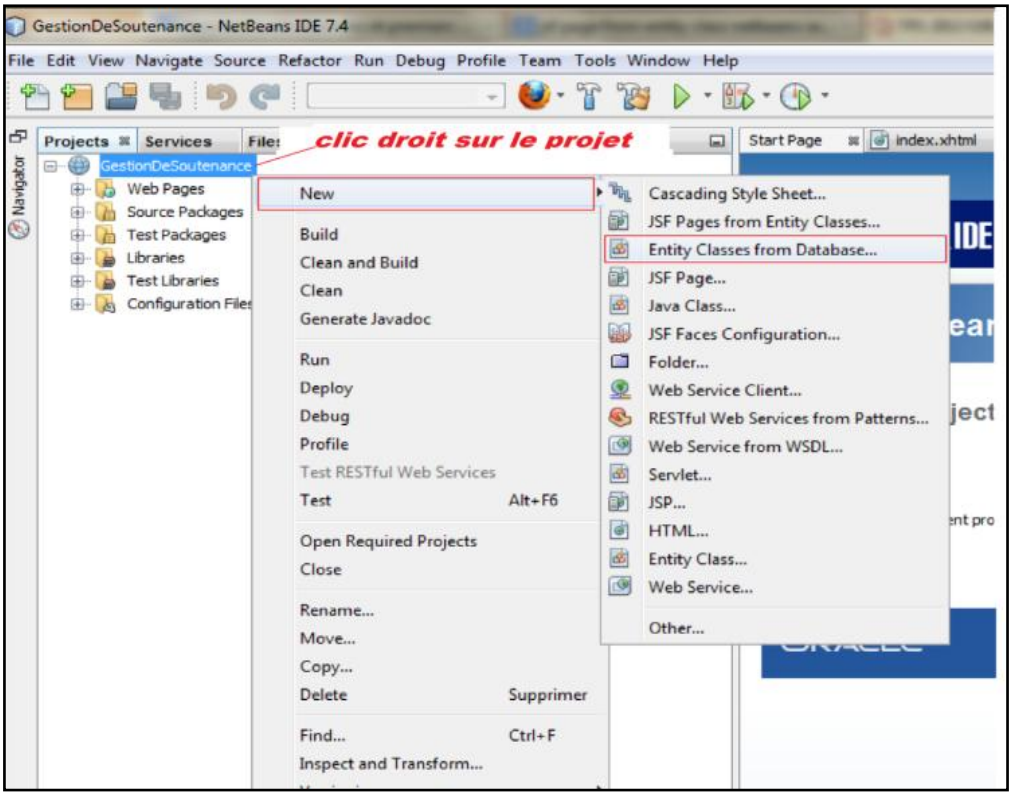


Figure 16 : Création des classes entités (étape 1)

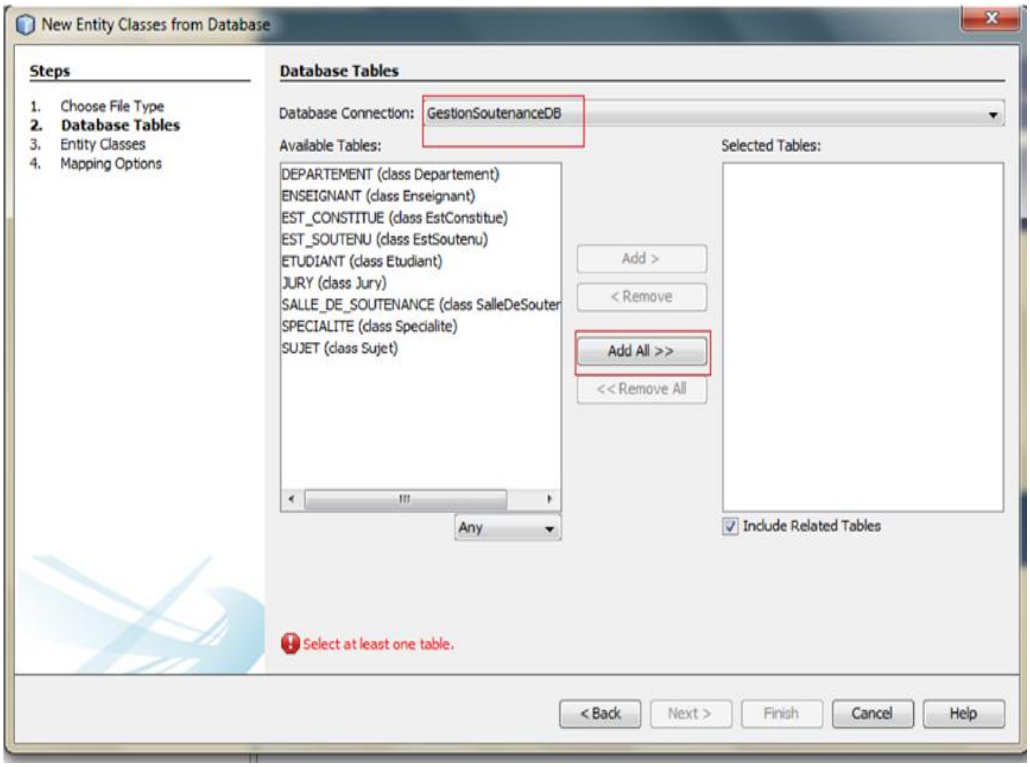


Figure 17 :Création des classes entités (étape 2)

Normalement dès que vous avez choisi le nom de la base, les tables s'affichent dans la colonne de gauche. Cliquez sur le bouton **"add all**", cela va ajouter en fait tous les tables.

- Cliquez sur le bouton **"Next"**. Maintenant on va vous proposer de changer le nom des classes entités correspondants à chacune des tables. Je conseille de laisser tout par défaut, cependant je conseille aussi de faire générer ces classes dans un package "entities" pour mieux structurer le projet :
- Cliquez sur le bouton "**Next**". L'étape suivante propose de modifier les valeurs par défaut pour les types de données correspondant à la relation 1-N entre les deux tables.
- Laissez tout par défaut et cliquez sur le bouton Finish.

Netbeans va un peu travailler puis vous verrez trois nouvelles choses dans le projet : 1-Les classes correspondant aux tables, dans le package "**entities**". 2- Un fichier **persistence.xml** sous "**configuration**", qui correspond à la définition d'une "**persistance unit**" par défaut. En réalité la persistance unit est l'objet qui va s'occuper de manipuler les données relationnelles à travers les classes et instances des entités. C'est la persistance unit qui connait la base, qui génère le SQL, qui va synchroniser les objets en mémoire et les données dans les tables, etc.

VI. Conclusion

Dans ce dernier chapitre, il s'agit d'affiner ce que nous avons vu tout au long du rapportet simplifier le travail en des interfaces qui résument la future application de la gestion d'emploi du temps des soutenances.

Conclusion Générale

Notre projet a été réalisé dans le cadre d'un projet de fin d'études et qui a pour objectif de développer une application de gestion d'emplois du temps des soutenances au sein de l'Université de Tlemcen.

Notre travail se résume en la conception et la réalisation d'une application web permettant la gestion des soutenances. 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'échangede 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 UML.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 Netbeans comme langage de programmation et Oracle comme système de gestion de base de données.

En outre, ce projet était une opportunité pour bien maitriser le développement web et apprendre le langage JAVA.

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 d'autres applications.

Bibliographie

[1] : https://aresu.dsi.cnrs.fr/spip.php?article95

[2] : https://aresu.dsi.cnrs.fr/spip.php?article95

[3] : https://aresu.dsi.cnrs.fr/spip.php?article95

[4] : https://aresu.dsi.cnrs.fr/spip.php?article95

[5] : https://aresu.dsi.cnrs.fr/spip.php?article95

[6] : https://aresu.dsi.cnrs.fr/spip.php?article95

[7] : Le processus unifié de développement logiciel

Édition : Eyrolles, juin 2000, ISBN10 : 2-212-09142-7

Auteur : Ivar Jacobson**,** Grady Booch**,** James Rumbaugh

[8] : http://sabricole.developpez.com/uml/tutoriel/unifiedProcess/

[9] : livre UML

[10] : http://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation

[11] : P. Roques et Franck Vallée, UML 2 en action, de l'analyse des besoins à la conception, Editions Eyrolles, 4e Edition]

[12] : http://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation

[13] : http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes

[14] : http://fr.wikipedia.org/wiki/Diagramme_de_sequence