



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

**Mise en place du catalogue de formation
détaillé de la faculté de sciences**

Réalisé par :

- **BENBOUBEKEUR Boumedine**
- **YOUSFI Houssam Eddine**

Présenté le 28 Mai 2015 devant la commission d'examen composée de MM.

- | | |
|----------------|-------------|
| - Messabihi M. | (Encadrent) |
| - Benazzouz M. | (Examineur) |
| - Mahfoud H. | (Examineur) |

Année universitaire : 2014-2015

Remerciements

Nous remercierons tout d'abord Dieu le tout puissant de nous avoir donné la santé, le courage et la patience pour pouvoir mener à bien ce travail.

En préambule à ce mémoire, on souhaiterait adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

On tient à remercier sincèrement monsieur Messabihi Mohamed, qui a accepté de nous encadrer et qui s'est toujours montré à l'écoute et très disponible et compétent tout au long de la réalisation de ce mémoire.

On tient à saluer les membres du jury qui nous ont honoré par leur présence et d'avoir accepté d'évaluer notre travail.

On n'oublie surtout pas nos parents pour leurs contributions, leurs soutiens et leur patience.

Enfin, on adresse nos plus chaleureux remerciements à tous nos proches et amis, qui nous ont toujours soutenus et encouragés au cours de la réalisation de ce mémoire.

Merci à toutes et à tous.

SOMMAIRE

Introduction générale.....	5
Chapitre 01 : GENERALITE.....	6
1.1 Présentation du système LMD.....	7
1.1.1 C'est quoi le système LMD ?.....	7
1.1.2 Les différents types des UE (Unité d'Enseignement).....	7
1.1.3 Les parcours de formation	8
1.1.4 Parcours académique ou professionnel ?.....	8
1.2 Terminologies.....	8
1.2.1 Crédit.....	8
1.2.2 Semestre.....	8
1.2.3 Année académique.....	9
1.3 Problématique.....	9
1.3.1 L'espace numérique de travail.....	9
1.3.2 Besoin d'un catalogue de formations.....	9
1.3.3 Situation actuelle.....	9
1.3.4 La solution proposée.....	10
2 CHAPITRE 02 : OUTILS UTILISÉS.....	11
2.1 Choix de la technologie la mieux adaptée à notre projet.....	12
2.1.1 Java Entreprise Édition (JEE).....	12
2.1.2 JSP (Java Server Pages Technology).....	12
2.1.3 SQL.....	13
2.1.4 HTML 5.....	13
2.1.5 CSS.....	13
2.1.6 JavaScript :.....	14
2.2 Choix des Outils.....	14
2.2.1 IDE Eclipse.....	14
2.2.2 MySQL WorkBench.....	14
2.2.3 UMLet.....	14
3 CHAPITRE 03 ANALYSE ET CONCEPTION.....	16
3.1 Analyse fonctionnelle.....	17
3.2 Analyse dynamique.....	18
3.2.1 S'authentifier.....	18
3.2.2 Rechercher une formation.....	19
3.2.3 Mettre à Jour.....	20
3.2.4 Supprimer.....	21
3.3 Modèle MVC.....	22
3.3.1 Modèle(Model VC).....	22
3.3.2 Vue (M View C)	23
3.3.3 Contrôleur (MV Controler) :.....	23
3.4 Conception structurelle:.....	24
4 CHAPITRE 04 : RÉALISATION	25
4.1 La structure de notre projet.....	26
4.2 Création de la base de données.....	26
4.3 Intégration de la BDD.....	28
4.3.1 JDBC	28
4.3.2 Data Access Object (DAO).....	29
4.3.3 Exemple d'une fonction dans le DAO :.....	29
4.3.3.1 Interaction avec la base de données.....	30
4.4 Présentation de notre catalogue de formation.....	31
4.4.1 Accès visiteur.....	31

4.4.1.1 La page d'accueil:.....	31
4.4.1.2 Navigation.....	32
4.4.1.3 Consulter la fiche technique de formation.....	32
4.4.1.4 La recherche d'une formation.....	33
4.4.2 Vue administrateur.....	34
4.4.2.1 L'insertion.....	34
4.4.2.2 La modification et la suppression	35
Conclusion générale.....	36
Bibliographie.....	37

Liste des figures

figure 1: système LMD	7
figure 2: Diagramme des cas d'utilisation.....	17
figure 3: Diagramme de séquence “ s'authentifier ”.....	18
figure 4: Diagramme de séquence “ Rechercher une formation ”.....	19
figure 5: Diagramme de séquence “Mettre à Jour”.....	20
figure 6: Diagramme de séquence “supprimer”.....	21
figure 7: Modèle MVC.....	22
figure 8: Diagrammes des classes du modèle.....	24
figure 9: L' hiérarchie du projet sous eclipse.....	26
figure 10: Modèle conceptuelle des données.....	27
figure 11: Une partie du scripte SQL de la création de la base de données.....	27
figure 12: Le modèle DAO.....	30
figure 13: La page d'accueil sur différente tailles d'écrans.....	32
figure 14: Listes de navigation.....	33
figure 15: Fiche semestrielle.....	33
figure 16: Détail d'un module.....	34
figure 17: La recherche.....	34
figure 18: Résultat d'une recherche.....	35
figure 19: Formulaire d'insertion d'une unité.....	36

Introduction générale

Dans le cadre de ce projet de fin d'étude et dans le cadre d'un projet de lancement d'un projet de création d'un Environnement Numérique de Travail (ENT) pour l'université de Tlemcen, nous avons voulu contribuer dans ce grand projet par la réalisation d'un module qui permet de gérer l'offre de formation à l'université de Tlemcen en créant un catalogue de formation en ligne. Il est intéressant de noter que l'université de Tlemcen (comme beaucoup d'autres université algériennes d'ailleurs) n'utilise pour le moment aucun système informatisé pour l'offre de formation.

Nous avons donc créé une application web que nous allons intégrer plus tard dans l'ENT déjà déployé par notre collègue Mr. Laidani Youssef étudiant en Master2 GL.

On a d'abord étudié les principes du système LMD et la nomenclatures des formations proposées dans le cadre de ce système. Et c'est ce que nous allons présenter dans le chapitre 1 de ce mémoire.

Ensuite, nous présenterons notre environnement de travail en donnant un aperçu sur les technologies utilisées au cours du développement, et en justifiant le choix de ces technologies.

Nous passerons par la suite à la conception de notre modèle de gestion de l'offre de formation, en détaillant les aspects fonctionnels, structurels et comportemental de notre système ainsi que ses interactions avec les autres module du grand projet de l'ENT.

Et enfin, nous terminerons par la présentation de notre solution, en soulignant quelques problématiques et les différentes solutions mises en œuvre.



Chapitre 01 : **GENERALITE**

Dans ce chapitre, nous allons commencer par présenter le système LMD, et plus particulièrement la structure des formations, ainsi que la terminologie utilisée dans ce nouveau système.

Ensuite, on parlera du catalogue de formation en tant que besoin réel pour la communauté universitaire.

1.1 Présentation du système LMD

L'Algérie a opté pour un nouveau système nommé LMD (Licence Master Doctorat) qui a remplacé le système classique.

Ce changement de l'enseignement à cause d'un mouvement dit mondialisation du système de formation universitaire[1].

1.1.1 C'est quoi le système LMD ?

Le LMD est un système de formation supérieure opté par le gouvernement algérien en 2004 .

- Une **architecture** des d'études en **3 grades**.
- Des **contenus structurés** en **domaines** comportant des **parcours types** et des **parcours individualisés**.
- Une **organisation** des **formations** en **semestres** et en **UE** (Unité d'Enseignement).

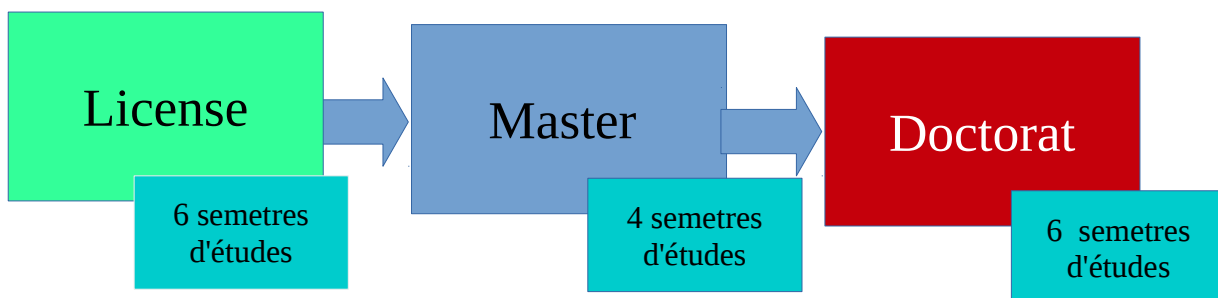


figure 1: système LMD

1.1.2 Les différents types des UE (Unité d'Enseignement)

Toute offre de formation comporte en général quatre grandes catégories d'UE agencées de manière pédagogique cohérente :

- **Les UE fondamentales (UEF)** : correspondant aux enseignements que tous les étudiants doivent suivre et valider.
- **Les UE méthodologiques (UEM)** : permettant à l'étudiant d'acquérir l'autonomie dans le travail.
- **Les UE découvertes (UED)** : permettant l'approfondissement, l'orientation, les passerelles, la professionnalisation....

- **Les UE transversales (UET)** : enseignements destinés à donner des outils aux étudiants : langue, informatique

1.1.3 Les parcours de formation

Il y a deux types de parcours :

- **Le parcours type** : est une combinaison cohérente d'UE constituant un cursus d'étude défini par l'équipe de formation et présenté dans l'offre de formation. Il est organisé de manière à permettre à l'étudiant d'élaborer progressivement son projet de formation.
- **Le parcours individualisés** : chaque étudiant peut construire un parcours « individualisé » avec l'aide et le suivi d'une ou plusieurs équipes de formation au sein d'un ou plusieurs établissements d'enseignement supérieure.

1.1.4 Parcours académique ou professionnel ?

Les deux parcours académique et professionnel devront s'adosser sur un même socle commun en première. Ils se différencient à partir du S3, d'où la licence professionnelle permettant une intégration directe dans le monde de travail et la licence académique ouvre la porte vers des études en Master

1.2 Terminologies

1.2.1 Crédit

L'UE et les matières qui composent le système LMD sont mesurées en « crédit » , Un Crédit correspond à la charge de travail (cours, stages, mémoire, travail personnel) requise pour l'étudiant atteigne les objectifs de l'UE ou de la matière.

Chaque semestre comporte 30 crédits. Chaque diplôme correspond à la capitalisation de :

- 180 crédits pour la licence.
- 120 crédits supplémentaires pour le Master.
- Le Doctorat s'obtient après 6 semestres d'études et de recherche.

1.2.2 Semestre

Le semestre est la durée périodique de l'enseignement , chaque semestre comporte un nombre fixe de semaines consacrées à l'enseignement et aux évaluations. Une moyenne de 14 à 16 semaines par semestre est raisonnable.

1.2.3 Année académique

Une année académique correspond à 60 crédits répartis à part égale sur deux semestres comme suit :

- 36 à 40 crédits sanctionnent des enseignements présentiels et travail personnel demandé
- 20 à 24 crédits restants sanctionnent les projets, les mémoires, les stages, ces crédits seront consacrés à l'enseignement présentiel et travail personnel de l'étudiant si l'année considérée ne prévoit pas de projets, stages , mémoires .
- Une année académique comprend un volume horaire approximatif.

1.3 Problématique

1.3.1 L'espace numérique de travail

L'espace numérique de travail, parfois aussi appelé cartable électronique est une plate-forme d'échanges qui rassemble tous les membres d'une communauté éducative d'un établissement scolaire ou universitaire, et leurs interlocuteurs au sein des académies et des collectivités.

L'E.N.T est un portail de services en ligne, c'est-à-dire un site web sécurisé, offrant un point d'accès unique où l'enseignant, l'élève et l'ensemble des personnels de l'établissement, peuvent trouver les informations, outils et services numériques en rapport avec leurs activités éducatives.

L'E.N.T est également destiné aux parents d'élève, qui peuvent trouver les informations concernant leur enfant et son établissement, ainsi que les moyens de communiquer avec le personnel éducatif.

Chaque utilisateur se connecte en se rendant sur une page web spécifique, sur laquelle il entre son nom et son mot de passe. Il accède ainsi à des services de base, personnalisés en rapport avec son activité[2].

1.3.2 Besoin d'un catalogue de formations

Un catalogue de formation c'est une maquette qui présente les grandes ligne d'une formations précise, sa durée, les objectifs visées, les prérequis ...

1.3.3 Situation actuelle

Dans la situation actuelle la seule façon pour consulter les formations d'une filière ou spécialité précise et d'attendre les portes ouvertes organisé par l'université chaque année après l'annonce des résultats de Bac pour les bacheliers et par chaque faculté pour les étudiants en L3. Malheureusement il n'y a pas de vrai catalogue de formation qui donne une vision globale sur les formations offertes par l'université surtout pour les nouveaux bacheliers ou leurs familles.

Inconvénients majeurs

- ➔ Nécessite de déplacer vers l'établissement concerné pour obtenir un catalogue de formations sous forme d'un dépliant (version papier).
- ➔ Beaucoup des déplacements parfois inutiles d'une ville à une autre.
- ➔ La lenteur dans l'accès aux informations.
- ➔ La difficulté de concevoir un parcours (type ou individuel).
- ➔ La contrainte de temps (temps limité).

1.3.4 La solution proposée

La solution proposée est l'informatisation de catalogue de formations sous forme d'une application web qui va apporter de nombreux avantages :

- ➔ Éviter le déplacement des étudiants pour se renseigner sur l'offre de formation.
- ➔ Accessibilités de l'information par un grand publique.
- ➔ Possibilités de concevoir son parcours type ou individuel.
- ➔ Diminuer la charge sur l'établissement lors des périodes des inscriptions.
- ➔ Faciliter l'accès aux informations.



2 CHAPITRE 02 : **OUTILS UTILISÉS**

Dans ce chapitre, nous citerons les langages et les technologies utilisés lors du développement de notre projet tout en justifiant notre choix.

2.1 Choix de la technologie la mieux adaptée à notre projet

En effet, plusieurs paramètres et critères devaient être considérés comme par exemples, les compétences déjà acquises. Dans notre cas, nous avions déjà une expérience non négligeable en programmation Java, nous avons donc considéré que JavaEE était la mieux adaptée à notre besoin et pourrait nous être très utile pour développer notre projet.

Parmi les avantages de l'utilisation de JEE, nous pouvons citer, entre autres :

1. une rapidité dans le développement
2. faible utilisation des ressources sur le serveur
3. réactivité de la communauté soutenant la technologie
4. la disponibilité de la documentation en ligne
5. la robustesse et la sécurité de l'application .

2.1.1 Java Enterprise Édition (JEE)



Java Enterprise Edition, ou Java EE (anciennement J2EE), est une spécification pour la technique Java d'Oracle 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 notamment la création d'applications réparties[3].

2.1.2 JSP (Java Server Pages Technology)



Le JavaServer Pages ou JSP est une technique basée sur Java qui permet aux développeurs de créer 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 conforme au standard 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 agissent 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. Il existe aussi un langage de script particulier, appelé Expression Language (EL) destiné à réduire l'injection de code java au sein des pages JSP ainsi qu'à étendre les possibilités des taglibs, tel que la JSTL.

Les JSP sont compilées par un compilateur JSP pour devenir des servlets Java. Un compilateur JSP peut créer une servlet Java en code source Java qui peut à son tour être compilé par le compilateur Java, ou peut créer le pseudo-code Java interprétable directement. Dans les deux cas, il est bon de comprendre comment le compilateur JSP transforme la page en servlet Java. Voir l'exemple de page JSP fourni en fin d'article, avec la servlet créée et la page HTML résultante [4].

2.1.3 SQL



Le **SQL** (Structured Query Language) est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web [5].

2.1.4 HTML 5

HTML



Cette spécification définit la 5ème révision majeure du langage de base du World Wide Web: Hypertext Markup Language (HTML). Dans cette version, de nouvelles fonctionnalités sont introduites pour aider les auteurs d'applications Web, de nouveaux éléments sont introduits basée sur la recherche dans les pratiques courantes de création, et une attention particulière a été accordée à la définition des critères de conformité claires pour les agents utilisateurs dans un effort pour améliorer l'interopérabilité[6].

2.1.5 CSS

CSS



Les feuilles de styles (en anglais "*Cascading Style Sheets*", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML ou XML.

Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fond, etc.

Le but de CSS est séparer la structure d'un document HTML et sa présentation. En effet, avec HTML, on peut définir à la fois la structure (le contenu et la hiérarchie entre les différentes parties d'un document) et la présentation [7].

2.1.6 JavaScript :



Le JavaScript est majoritairement utilisé sur Internet, conjointement avec les pages Web HTML. Le JavaScript s'inclut directement dans la page Web (ou dans un fichier externe) et permet de *dynamiser* une page HTML, en ajoutant des interactions avec l'utilisateur, des animations, de l'aide à la navigation, comme par exemple :

- Afficher/masquer du texte ;
- Faire défiler des images ;
- Créer un diaporama avec un aperçu « en grand » des images ;
- Créer des infobulles.

Le JavaScript est un langage dit *client-side*, c'est-à-dire que les scripts sont exécutés par le navigateur chez l'internaute (le **client**). Cela diffère des langages de scripts dits *server-side* qui sont exécutés par le serveur Web[8].

2.2 Choix des Outils

2.2.1 IDE Eclipse



L'environnement de développement de référence pour Java/Java EE. Il supporte un grand nombre de plugins qui le rendent utilisable pour d'autres langages et d'autres technologies.

Eclipse est une plateforme de développement écrite en Java, fruit du travail d'un consortium de grandes entreprises (IBM, Borland, Rational Rose, HP...). Il en résulte un IDE performant et Open Source qui a su trouver sa place comme l'un des environnements de développement Java les plus populaires. Elle intègre pour cela la prise en charge des outils comme Ant, SVN, JUnit... [9].

2.2.2 MySQL WorkBench



MySQL Workbench est un outil visuel unifié pour les architectes de bases de données, les développeurs et les administrateurs de bases de données. MySQL Workbench propose une modélisation de données, le développement de SQL et les outils d'administration complets pour la configuration du serveur, gestion des utilisateurs, la sauvegarde, et bien plus encore. MySQL Workbench est disponible sous Windows, Linux et Mac OS X [10].

2.2.3 UMLet



UMLet est un logiciel open-source avec une simple interface d'utilisateur, permet de dessiner des diagrammes UML facilement, et il peut aussi

construire des diagrammes à partir d'une description textuelle. Il permet également d'exporter les diagrammes UML dans différents formats [11].



3 CHAPITRE 03

ANALYSE ET

CONCEPTION

L'analyse et la conception de l'application est une phase cruciale et très importante pour réaliser notre projet. En effet, si les besoins sont mal spécifiés et exprimés, ou mal analysés, le projet ne serait pas réalisé correctement, d'où l'importance que nous avons accordée à cette étape.

Pour la conception de notre projet, nous avons choisi d'utiliser le langage UML pour détailler les aspects fonctionnels, dynamiques et structurels.

3.1 Analyse fonctionnelle

Le diagramme ci-dessous représente les différentes cas d'utilisations (USE CASES), il aide à décomposer le système en plusieurs fonctionnalités.

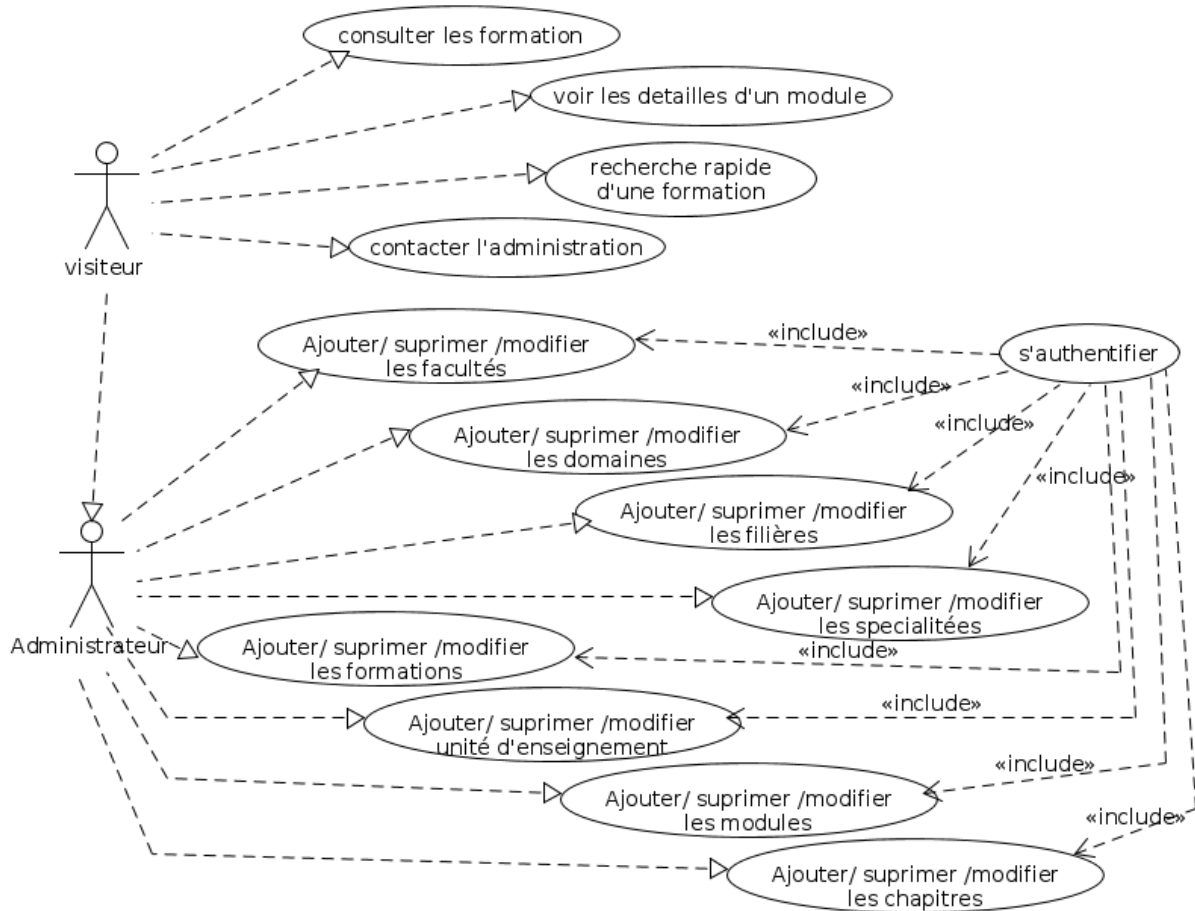


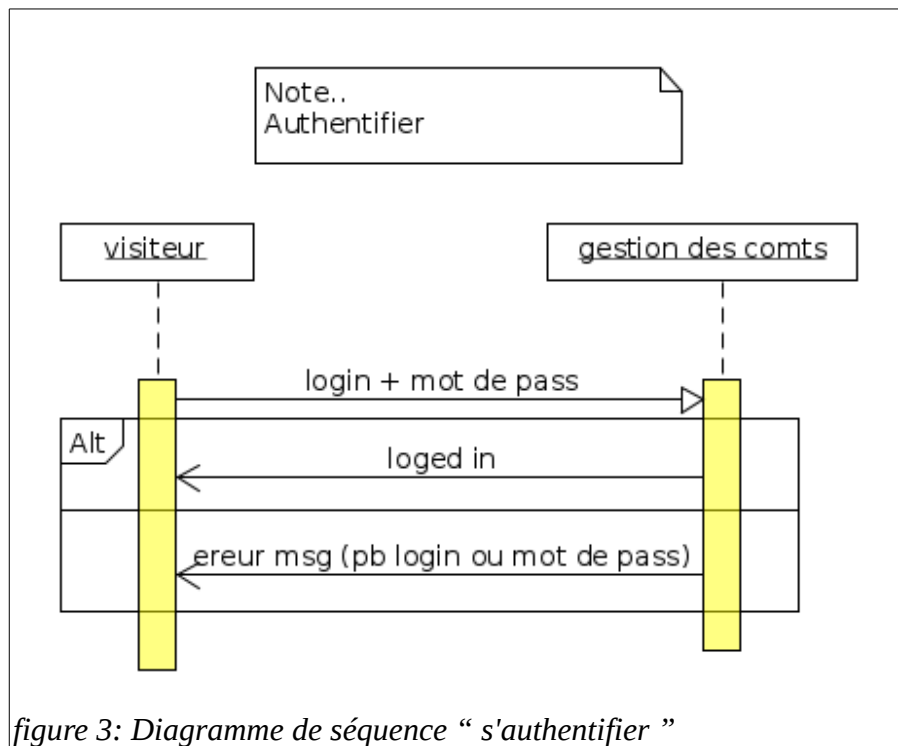
figure 2: Diagramme des cas d'utilisation

On distingue deux acteurs, un simple visiteur qui peut consulter le site, et un administrateur qui peut modifier le contenu du site. L'administrateur doit être authentifié.

3.2 Analyse dynamique

3.2.1 S'authentifier

L'authentification permet à l'administrateur de saisir son login et son mot de passe. Elle est utilisée dans le système car c'est une manière de garder la sécurité et la cohérence de système.



Le diagramme ci-dessus décrit les interactions entre l'administrateur et le système lors de l'authentification.

3.2.2 Rechercher une formation

On offre à l'utilisateur deux mécanismes de recherche (voire la figure 4):

1. le premier est une simple navigation à travers des menus dans la page principale de l'application
2. le deuxième est un champ de recherche rapide où l'utilisateur saisie le mot clé désiré ou une partie de ce mot (lazy search).

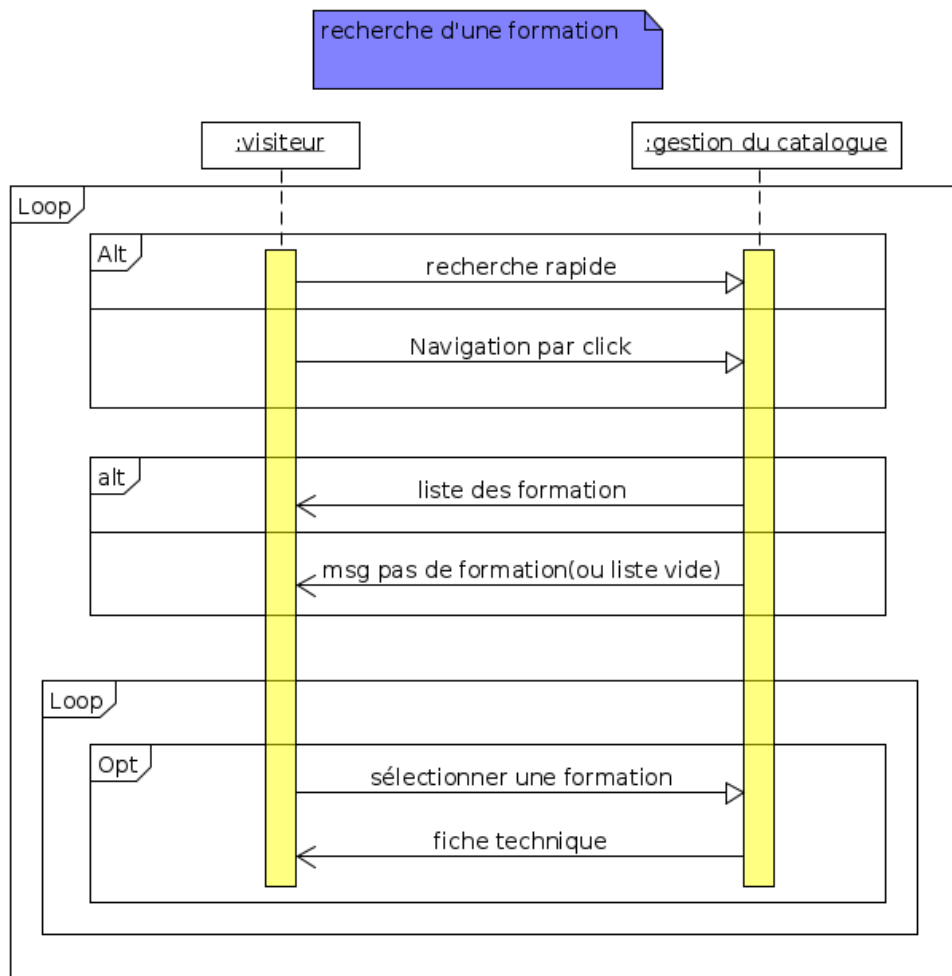


figure 4: Diagramme de séquence “ Rechercher une formation ”

Le système retourne une liste de formations qui correspondent au mot clé recherché, sinon une liste vide. Dans la page des résultats, l'utilisateur peut naviguer à travers des liens vers d'autres pages pour obtenir plus de détails.

3.2.3 Mettre à Jour

Dans le système, il existe plusieurs opérations d'ajout/modification selon les différents types des informations et les tables qui existent dans la BDD (spécialités, modules...), mais toutes ces opérations partagent la même procédure représentée dans le diagramme ci-dessus.

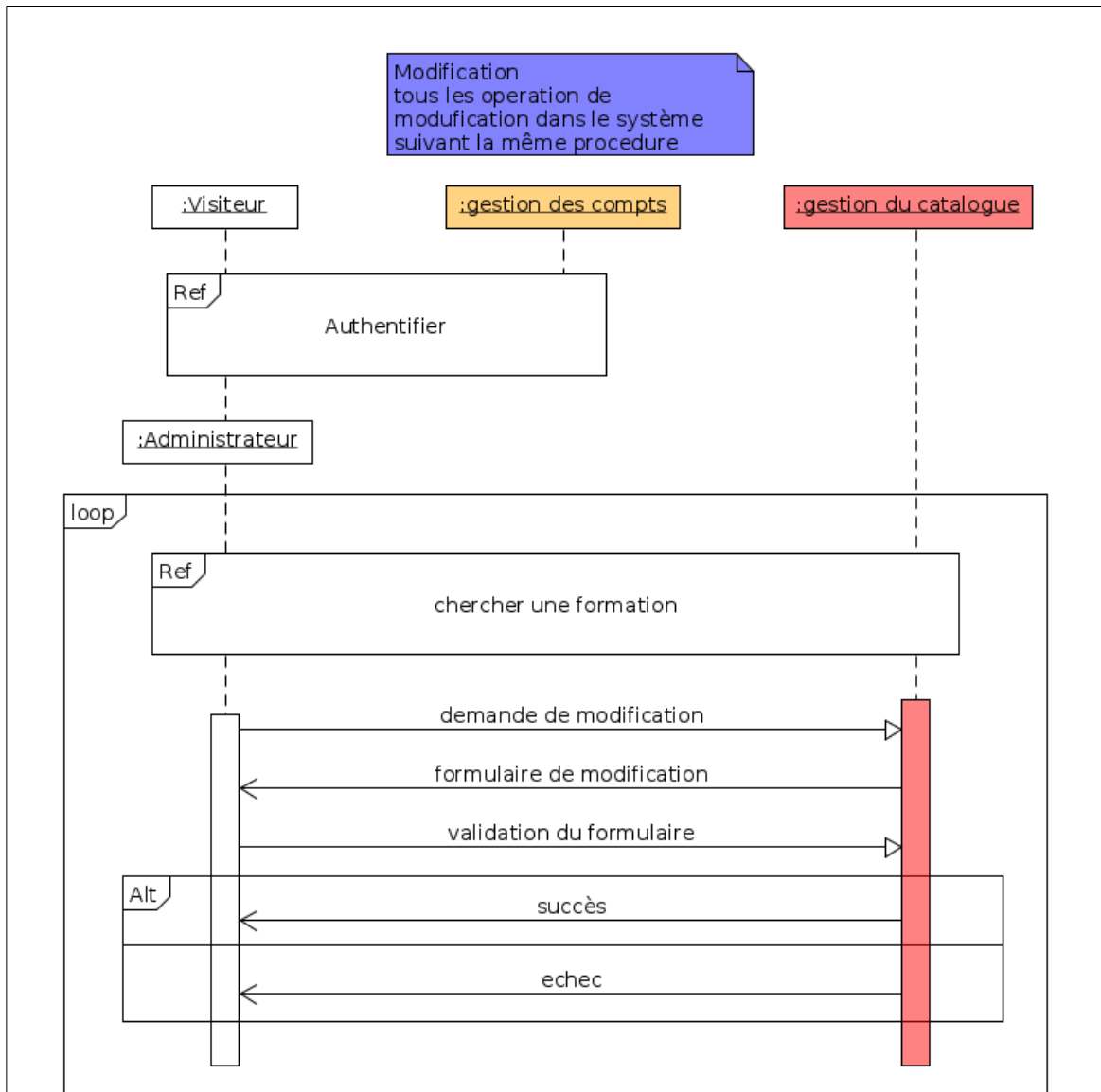


figure 5: Diagramme de séquence "Mettre à Jour"

L'administrateur doit s'authentifier afin de confirmer qu'il a le droit de modifier. Il remplit le formulaire puis valide ces modifications. Le système répond par message de succès ou d'échec.

3.2.4 Supprimer

De même pour l'ajout/ modification, la suppression dépend de type d'information à supprimer, et elle nécessite l'authentification aussi, mais elle consiste simplement à trouver cette information, et demander sa suppression.

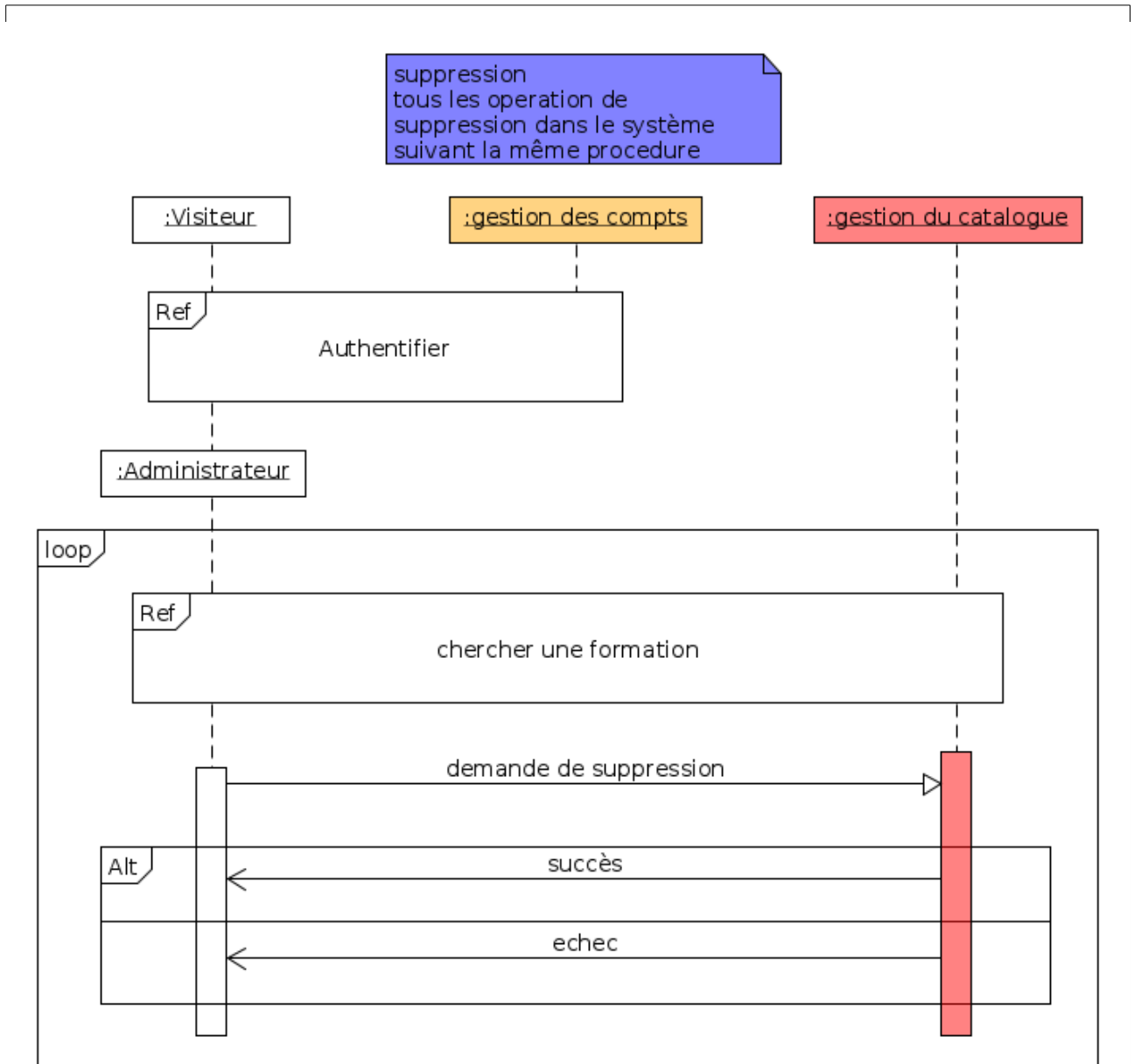


figure 6: Diagramme de séquence "supprimer"

3.3 Modèle MVC

- Un modèle de conception est une simple bonne pratique, qui répond
- à un problème de conception d'une application. C'est en quelque sorte un guide qui permet de décrire les grandes lignes d'une solution dans notre cas on a optez le modèle MVC.
- Cet modèle découpe littéralement l'application en couches distinctes, et de ce fait impacte très fortement l'organisation du code ! les grandes lignes ce modèle :
- tout ce qui concerne le traitement, le stockage et la mise à jour des données de l'application doit être contenu dans la couche nommée « Modèle » (le M de MVC) ;
- tout ce qui concerne l'interaction avec l'utilisateur et la présentation des données (affichage) doit être contenu dans la couche nommée « Vue » (le V de MVC) ;
- tout ce qui concerne le contrôle des actions de l'utilisateur et des données doit être contenu dans la couche nommée « Contrôle » (le C de MVC)[12].

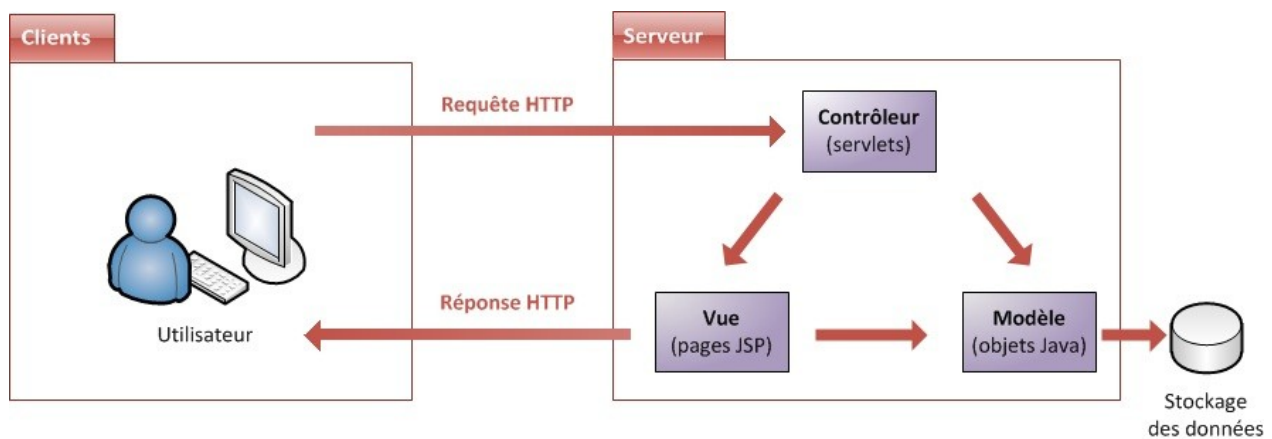


figure 7: Modèle MVC

3.3.1 Modèle(Model VC)

Dans le modèle, on trouve à la fois les données et les traitements à appliquer à ces données. Les données sont représentées temporairement par des « beans » qui sont des simples classes Java au niveau de notre programme et représentées de manière permanente avec des tables au niveau de BDD manipulable à travers un système capable d'effectuer les tâches suivantes CRUD (Creute Read Update Delete) les données qui est dans notre cas un DAO (Data Access Object).

Ce DAO contient toutes les implémentations des méthodes nécessaires pour que nous puissions faire les tâches précédentes et plus.

Pour le traitement des données on a utilisé des classes nommées « Formes » qui ont pour but principal de traiter les données reçues de l'utilisateur et d'invoquer les méthodes de CRUD nécessaires.

3.3.2 Vue (M View C)

Cette couche nommée « Vue » (le V de MVC) est concernée par l'affichage et les représentations des données aux utilisateurs. C'est à travers cette couche qu'on peut interagir avec l'utilisateur.

Dans notre projet, on a utilisé la technologie JSP (Java Server Page) définie précédemment qui permet de générer des pages HTML dynamiques.

3.3.3 Contrôleur (MV Controller) :

Le contrôleur est une brique essentielle dans le modèle MVC, il joue le rôle d'un aiguilleur entre la Vue et le Modèle et il est implémenté par des servlets dans notre application.

Le client envoie des requêtes HTTP (get(), post()) et notre servlet se contente d'appeler les traitements modèles appropriés et finalement rendre le résultat au client sous forme d'une page HTML générée par la Vue.

3.4 Conception structurelle:

Le diagramme de classes représente, en générale, la structure d'une application, ce diagramme contient uniquement les entités (la partie modèle de notre application).

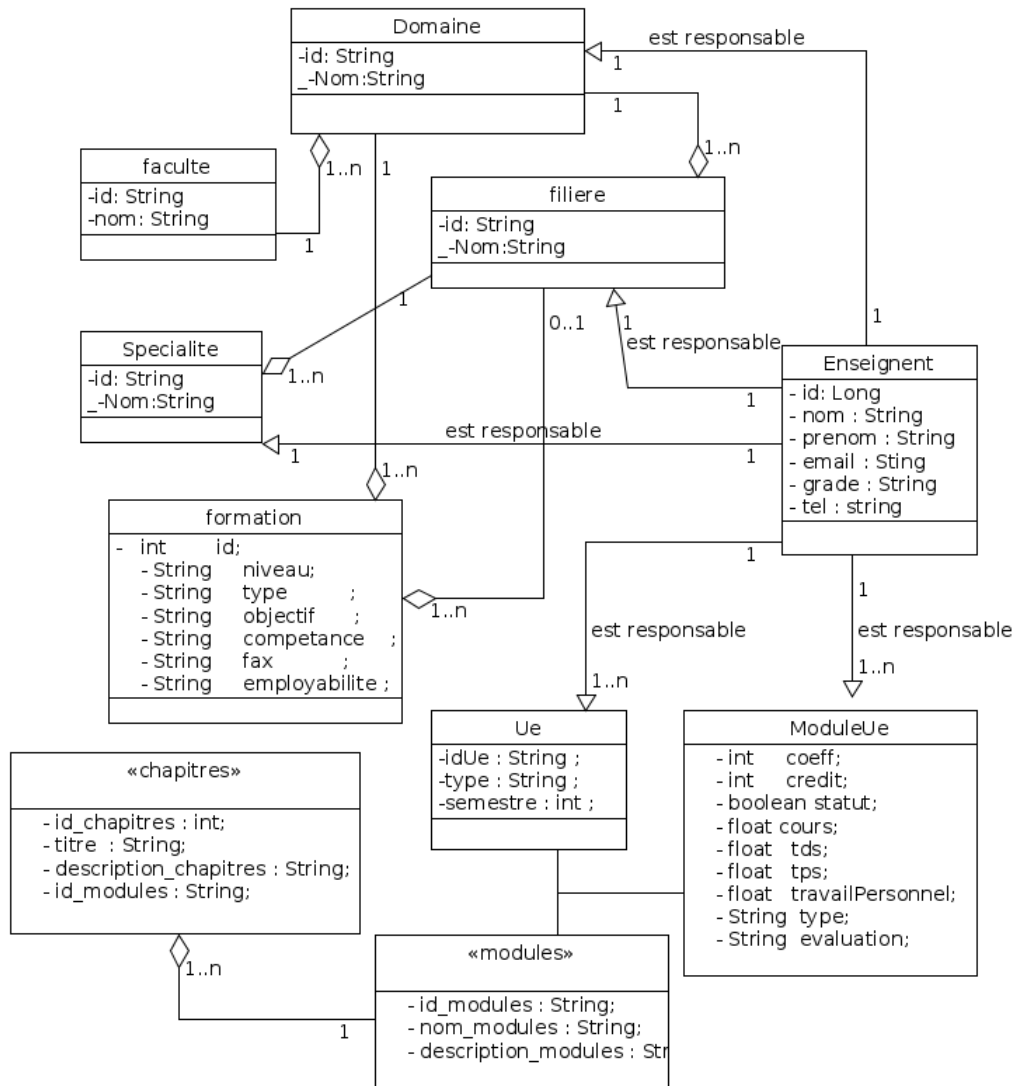


figure 8: Diagrammes des classes du modèle

On peut voir clairement dans le diagramme ci-dessus une série d'agrégations (Domaine >> Filières >> Spécialités >> ... ect) car en réalité c'est une hiérarchie.

Il existe d'autre relation dans le diagramme comme pare exemple la relation « est responsable » (une relation entre un enseignant et spécialité ou UE...).



4 CHAPITRE 04 : **RÉALISATION**

Après avoir terminé l'étape de conception de notre application, nous abordons dans ce chapitre la réalisation de notre application qui constitue la dernière étape de cette étude.

4.1 La structure de notre projet

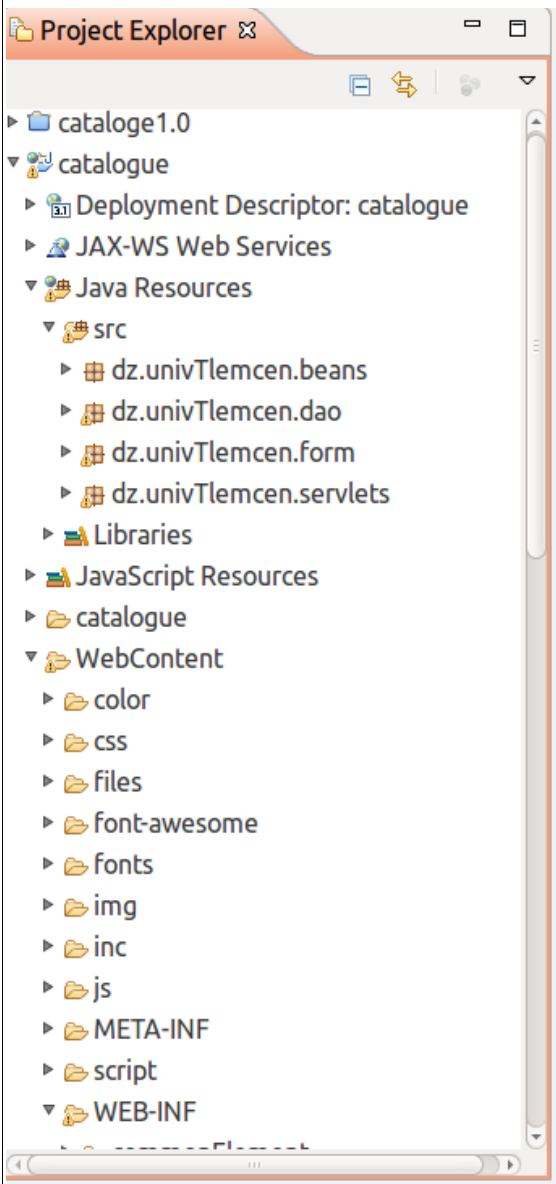


figure 9: L' hiérarchie du projet sous eclipse

- Le dossier qui porte le nom «catalogue» est la racine de projet.
- le dossier nommé «Web Content», contient un dossier src qui contiendra le code source des classes (les fichiers.java), ces classe son répartie en 4 packages.
- Le dossier nommé WEB-INF est un dossier spécial. Il doit obligatoirement exister et être placé juste sous la racine.
- le fichier de configuration de l'application (web.xml) se trouve dans WEB-INF;
- Le dossier nommé classes contient les classes compilées (fichiers .class) ;
- Le dossier nommé lib contient les bibliothèques nécessaires au projet (archives .jar).

4.2 Création de la base de données

Avant de créer une BDD il fallait passer par la phase d'analyse et l'élaboration d'un modèle conceptuel de données, c'est pourquoi nous avons utilisé MySQL WorkBench pour effectuer cette tâche. Le résultat est donné dans la figure ci-dessous.

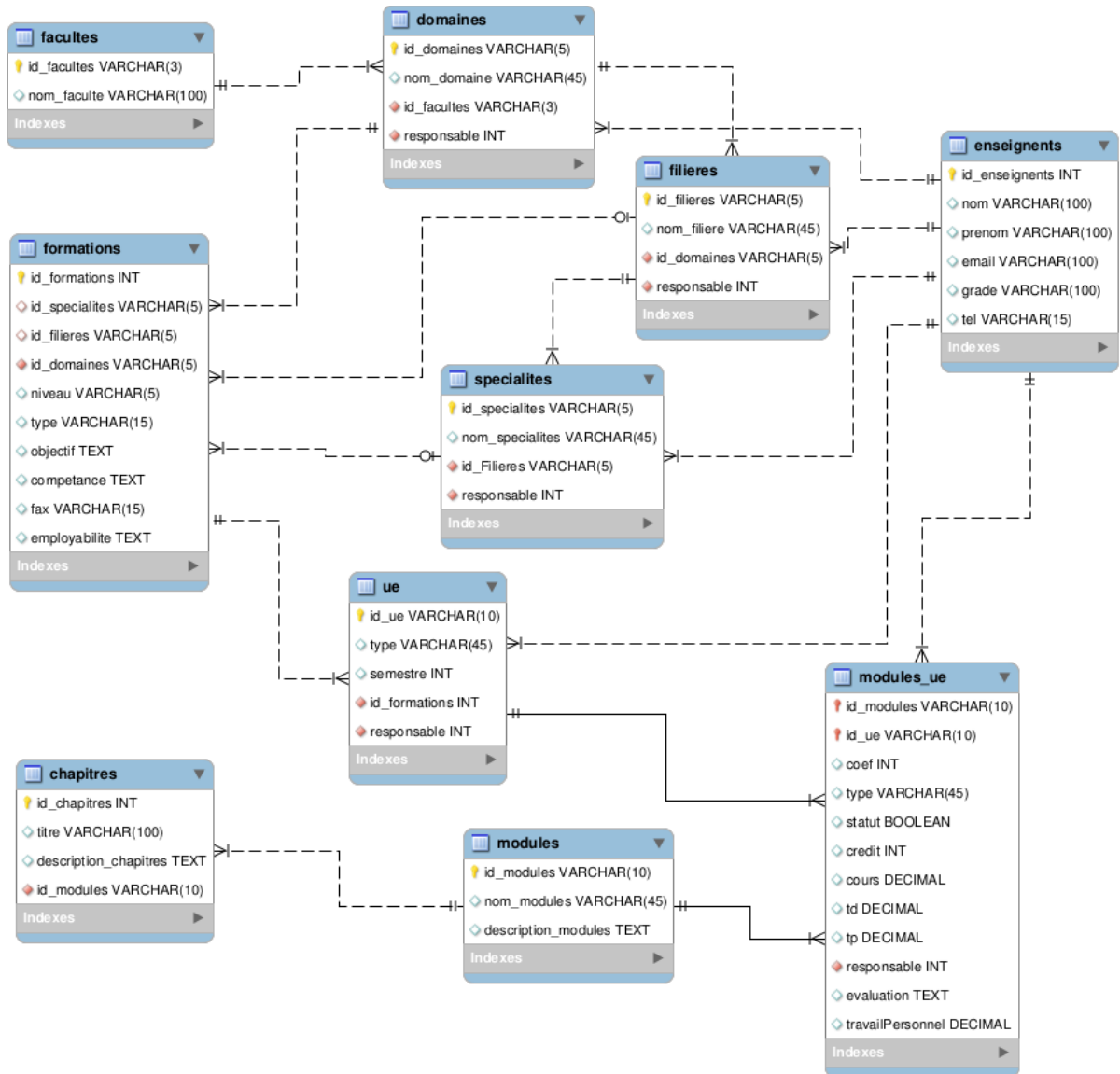


figure 10: Modèle conceptuelle des données

Nous avons utilisé le même outils pour générer le script SQL à exécuter pour créer notre base de données. Un extrait de ce script est donné ci-dessous :

```

-----
-- Schema formation
-----
CREATE SCHEMA IF NOT EXISTS `formation` DEFAULT CHARACTER SET utf8 ;
USE `formation` ;

-----
-- Table `formation`.`modules`
-----
CREATE TABLE IF NOT EXISTS `formation`.`modules` (
  `id_modules` VARCHAR(10) NOT NULL,
  `nom_modules` VARCHAR(45) NULL DEFAULT NULL,
  `description_modules` TEXT NULL DEFAULT NULL,
  PRIMARY KEY (`id_modules`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
    
```

figure 11: Une partie du scripte SQL de la création de la base de données

4.3 Intégration de la BDD

La phase de création d'une BDD est terminée, mais notre travail est loin d'être fini. La prochaine étape est d'établir la connexion entre MySQL et notre projet JavaEE dans l'objectif de pouvoir interagir avec les tables de notre BDD directement depuis le code de l'application. Pour pouvoir faire cela, nous avons utilisé JDBC (Java DataBase Connector).

4.3.1 JDBC

Le JDBC (Java DataBase Connector) est une API qui fait partie intégrante de la plate-forme Java, et qui est constituée de classes permettant l'accès depuis des applications Java à des données rangées sous forme de tables. Cette API fournit plusieurs actions qui sont :

- ➔ la connexion avec le SGBD.
- ➔ l'envoi de requêtes SQL au SGBD depuis une application Java.
- ➔ le traitement des données et éventuelles erreurs retournées par le SGBD.

La figure suivante montre une méthode pour charger le driver :

Le figure suivante montre une méthode pour charger le driver :

```
19
20 public static DaoFactory getInstance()
21 {
22     // chargement du driver
23     try
24     {
25         Class.forName( "com.mysql.jdbc.Driver" );
26     } catch ( ClassNotFoundException e )
27     {
28     }
29
30     // cree l'objet factory
31     DaoFactory instance = new DaoFactory(
32         "jdbc:mysql://localhost:3306/formation", "root", "" );
33     return instance;
34 }
```

Implémentation 1: Chargement du driver JDBC

La figure suivante montre la méthode pour établir la connexion :

```
36 // fonction pour etablire la connexion
37 public Connection getConnection() throws SQLException
38 {
39     return DriverManager.getConnection( url, userName, passWord );
40 }
41
```

Implémentation 2: Établissement d'une connexion à la base des donnée

4.3.2 Data Access Object (DAO)

Le design paterne DAO est une pratique recommandée par les développeurs expérimentés, car c'est une extension du modèle MVC, et il permet de mieux structurer notre code, parmi les avantages de l'utilisation des DAO nous citons :

- La facilité de la maintenance de l'application.
- La possibilité de l'extension.
- La facilités de changer le SGBD ou modifier la BDD existante.
- Et finalement, la possibilité de tester l'application sans BDD.

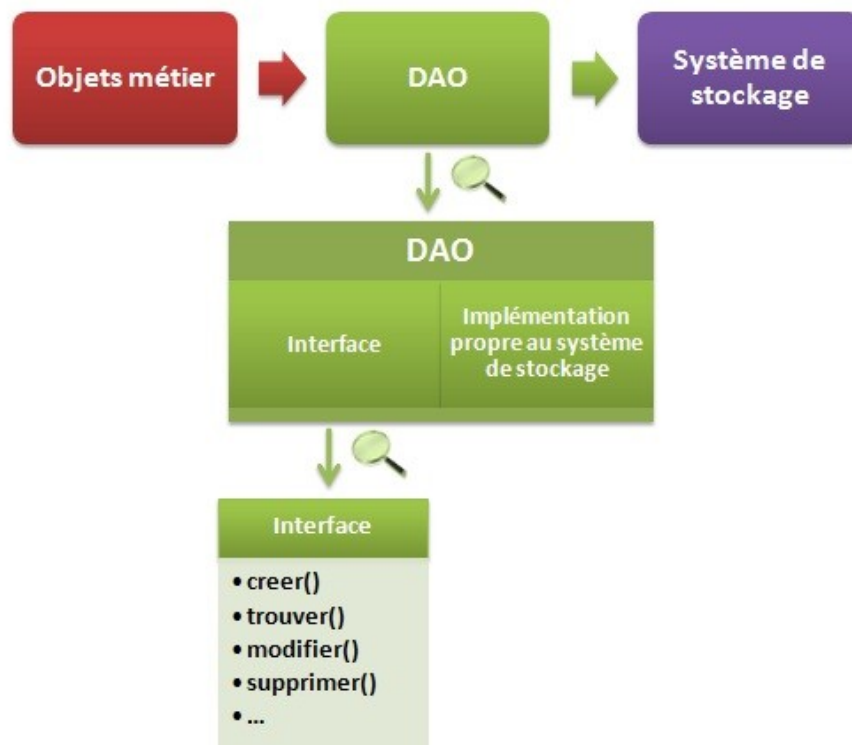


figure 13: Le modèle DAO

Les DAO sont des interfaces Java avec une implémentation qui varie selon le SGBD utilisé [11].

4.3.3 Exemple d'une fonction dans le DAO :

Pour cet exemple on a choisi une fonction qui retourne une liste de toutes les formations. Cette opération s'exécute en trois étapes :

1. La première étape et d'établir une connexion avec la base des données en utilisant

l'objet «connection».

2. Ensuite préparer la requête avec un objet «statement» en utilisant la méthode `prepareStatement()` qui permet d'éviter la faille SQL injection.
3. Enfin, on utilise la fonction `executeQuery()`, qui retourne une liste chaînée (un objet `resultSet`) qu'on va ensuite transformer sous forme d'un `arrayListe` pour faciliter la manipulation des données dans la vue.

Pour finir, on capture les exceptions et on ferme la connexion.

```
public ArrayList<Formation> getAllFormations() {
    ArrayList<Formation> allFormations = new ArrayList<Formation>();
    // TODO Auto-generated method stub

    Connection connection = null;
    PreparedStatement statement = null;
    ResultSet brutResult = null;

    try {
        connection = daoFactory.getConnection();
        statement = connection.prepareStatement(SQL_SELECT_ALL_FORMATIONS);
        brutResult = statement.executeQuery();

        while (brutResult.next()) {
            Formation formation = new Formation();
            formation.setId(brutResult.getInt(COLUMN_ID_FORMATION));
            formation.setSpecialite(specialiteDao
                .getSpecialiteById(brutResult
                    .getString(COLUMN_ID_SPECIALITE)));
            formation.setDomaine(domaineDao.getDomaineById(brutResult
                .getString(COLUMN_ID_DOMAINE)));
            formation.setFiliere(filiereDao.getFiliereById(brutResult
                .getString(COLUMN_ID_FILIERE)));
            formation.setNiveau(brutResult.getString(COLUMN_NIVEAU));
            formation.setType(brutResult.getString(COLUMN_TYPE));
            formation.setFax(brutResult.getString(COLUMN_FAX));
            formation.setObjectif(brutResult.getString(COLUMN_OBJECTIFS));
            formation
                .setCompetance(brutResult.getString(COLUMN_COMPETANCE));
            formation.setEmployabilite(brutResult
                .getString(COLUMN_EMPLOYABILITE));

            allFormations.add(formation);
        }
    } catch (SQLException e) {
    } finally {
        fermetureConnection.fermetures(brutResult, statement, connection);
    }
}
```

Implémentation 3: Récupération de la liste de tout les formation

4.3.3.1 Interaction avec la base de données

Afin de montrer comment notre site interagit avec la base de données, dans cette section, nous présentons une fonction qui s'occupe de la recherche d'une formation dans toute

la base des données.

```

SELECT *
FROM formations
WHERE id_filières in(
    SELECT id_filières
    From filieres
    WHERE nom_filiere LIKE ? )
OR id_domaines in(
    SELECT id_domaines
    From domaines
    WHERE nom_domaine LIKE ?)
OR id_specialites in(
    SELECT id_specialites
    From specialites
    WHERE nom_specialites LIKE ?)
OR id_formation in(
    SELECT ue.id_formation
    From ue,modules_ue,modules
    WHERE ue.id_ue = modules_ue.id_ue
    AND modules_ue.id_modules=modules.id_modules
    AND modules.nom_modules LIKE ? );

```

Implémentation 4: Requête SQL de la recherche

4.4 Présentation de notre catalogue de formation

Comme indiqué dans le diagramme de cas d'utilisation, on distingue deux acteurs principaux; le visiteur et l'administrateur. Dans la suite de ce chapitre, nous détaillons la vue correspondant à chacun de ces deux profils.

4.4.1 Accès visiteur

4.4.1.1 La page d'accueil:

On a utilisé le « responsive view » dans le design de notre Application c'est à dire l'interface de notre site change selon la taille d'écran ce qui donne :

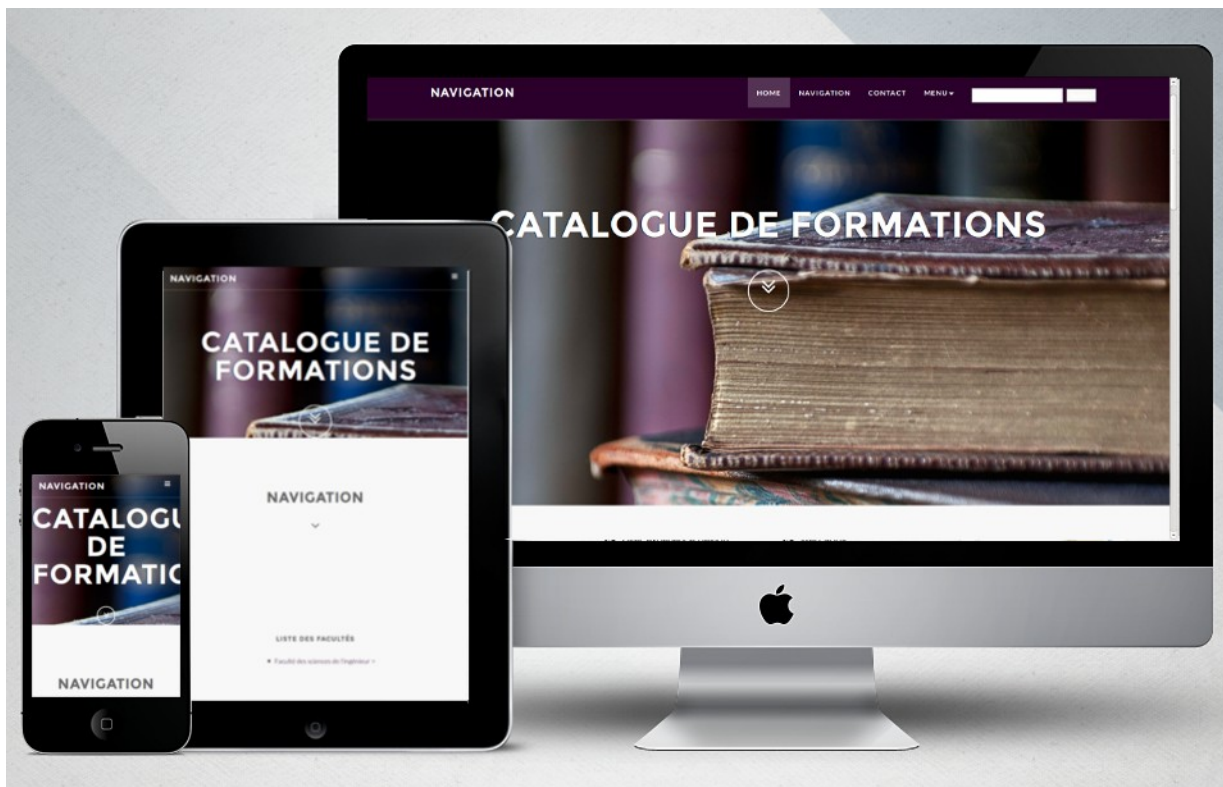


figure 13: La page d'accueil sur différentes tailles d'écrans

4.4.1.2 Navigation

Sous la bannière du site on trouve directement une section de navigation où l'utilisateur peut choisir une formation simplement par une cascade de listes devant lui :

LISTE DES FACULTÉS	LISTE DES DOMAINES
<ul style="list-style-type: none"> • Facultés des langues étrangères > • Facultés de médecine > • Faculté des sciences > • Facultés des technologie > 	<ul style="list-style-type: none"> • Mathématiques, Informatique > • Sciences de la Matière > • science technique >
LISTE DES FILIERES	LISTE DES FORMATIONS
<ul style="list-style-type: none"> • Informatique > • Mathématiques > 	<ul style="list-style-type: none"> • Informatique L1 • Informatique L2 • Informatique L3 • Génie logiciel M1 • Génie logiciel M2 • Réseaux et Systèmes Distribués M1 • Réseaux et Systèmes Distribués M2 • Modèles intelligents et décision M1 • Modèles intelligents et décision M2 • Systèmes d'information et de connaissance M1 • Systèmes d'information et de connaissance M2

figure 14: Listes de navigation

4.4.1.3 Consulter la fiche technique de formation

Après avoir choisi une formation précise, l'utilisateur trouve une fiche technique avec les informations utiles telle que le contexte et les objectifs de la formation, les coordonnateurs ainsi que les fiches semestrielles (voir figure 15).

1- SEMESTRE 1 :

Unité d'Enseignement	V.H hebdomadaire				Coef	Crédits	Mode d'évaluation
	C	TD	TP	Autre			
UE : Rx11 Fondamental							
Algorithmique avancée et complexité	1.5	1.5	1.5	3.0	3.0	5.0	50% Continu 50% Examen
Modélisation et évaluation des performances des systèmes	1.5	1.5	0.0	3.0	4.0	5.0	50% Continu 50% Examen
UE : Rx12 Fondamental							
Gestion de projets informatiques	1.5	0.0	1.5	3.0	5.0	5.0	34% Continu 66% Examen
Etudes et administration des SGBD	1.5	0.0	1.5	3.0	5.0	5.0	34% Continu 66% Examen
UE : Rx13 Fondamental							
Programmation Concurrente et temps réel	1.5	0.0	3.0	3.0	3.0	4.0	66% Continu 34% Examen
Réseaux II	1.5	1.5	1.5	3.0	5.0	5.0	66% Continu 34% Examen

figure 15: Fiche semestrielle

L'utilisateur peut obtenir plus de détails sur chaque module, par exemple s'il clique sur le module nommée " Génie logiciel " il obtiendra l'affichage suivant :

The screenshot shows a web page titled "DÉTAILLE DU MODULE" with a navigation bar containing "HOME", "NAVIGATION", "CONTACT", and "MENU". A search bar is visible with the text "Recherche" and a "Go" button. The main content is divided into two columns. The left column contains a table with the following data:

Intitulé de la matière	Génie logiciel
Code	GL111
Semestre	1
Unité d'Enseignement	GL12 Fondamental
Enseignant responsable de la matière:	CHIKH Mohammed Amine
Enseignant responsable de l'UE:	CHIKH Mohammed Amine
Nombre d'heures d'enseignement	Cours :1.5 TD :1.5 TP :0.0
Nombre d'heures de travail personnel pour l'étudiant	3.0
Coefficient	3.0
Credit	4.0 crédits
Type Module	Oblogatoire
Statut	Active

The right column contains the following sections:

DESCRIPTION
L'objectif de ce module consiste à illustrer les concepts du GL pour le développement d'un produit informatique (logiciel) de qualité. Il concerne différentes thématiques : de la gestion de projet (cycles de développement), réalisation des spécifications à l'aide de méthode formelle, etc. Un projet en java est à réaliser par équipe.

CONTENU

- Les problèmes du développement logiciel
- Le cycle de vie
- Méthodes et modèles
- La spécification
- La conception
- La programmation
- Le test
- L'assurance qualité logiciel

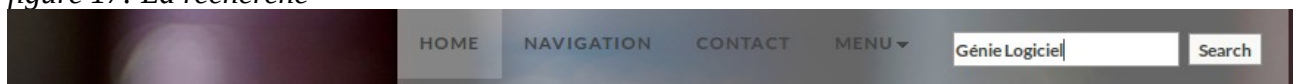
figure 16: Détail d'un module

4.4.1.4 La recherche d'une formation

L'utilisateur n'est pas obligé d'utiliser le menu de navigation pour retrouver une formation, car il peut effectuer une recherche rapide à tout moment à l'aide d'un champ de recherche présent toujours en haut de la page.

Dans cet exemple l'utilisateur a saisi le mot clé " Génie Logiciel ".

figure 17: La recherche



Le résultat de la recherche est une liste des formations qui ont une relation avec le mot clé entré par l'utilisateur, dans cet exemple le résultat est le suivant.

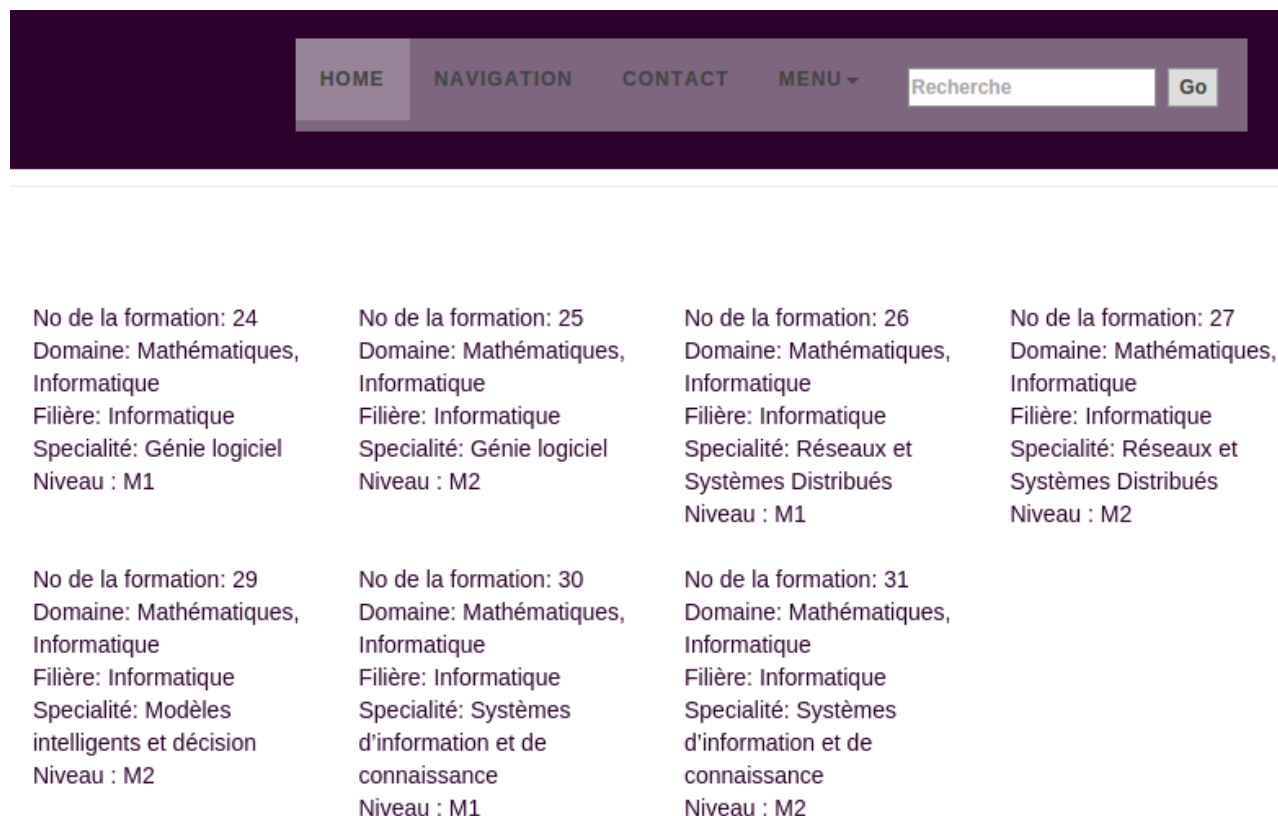


figure 18: Résultat d'une recherche

4.4.2 Vue administrateur

Pour la partie administrateur l'interface est simple et efficace, car la même interface est utilisée dans tous les modules administratifs, pour que l'administrateur se familiarise avec ce mode d'utilisation. Cette interface offre les opérations suivantes :

4.4.2.1 L'insertion

Les opérations d'insertion consistent à remplir un formulaire et de valider les données saisies. Pour illustrer l'insertion, la figure suivante montre comment insérer une nouvelle Unité d'enseignement.

Unité d'Enseignement:

ID de l'unité d'enseignement

Type du UE

Semestre

Formation

Responsable

figure 19: Formulaire d'insertion d'une unité

4.4.2.2 La modification et la suppression

Pour ces deux opérations, il suffit d'appuyer sur un bouton, la suppression se fait directement, et la modification renvoie vers un formulaire similaire à l'insertion.

Rx11	Fondamental	1	MI RSD M1	Benazzouz Mortada	<input type="button" value="Supprimer"/>	<input type="button" value="modifier"/>
Rx13	Fondamental	1	MI RSD M1	Benaissa Mohamed	<input type="button" value="Supprimer"/>	<input type="button" value="modifier"/>
Rx31	Fondamental	1	MI RSD M2	Benallel Mounira	<input type="button" value="Supprimer"/>	<input type="button" value="modifier"/>

Conclusion générale

Au vu de ce qui précède, nous pouvons retenir que le développement d'une application web en JAVA/J2EE pour la réalisation d'une application de gestion catalogue des formations nécessite essentiellement toute une étude approfondie avec de nombreuses approches méthodologiques d'analyse et de conception, et des outils de modélisation de données et de développement.

Dans ce modeste rapport de projet de fin d'étude, nous avons présenté, en quatre parties, l'essentiel des étapes à suivre pour atteindre les objectifs souhaités. Nous avons réalisé une application sous les normes qui a la possibilité de gérer efficacement et surtout simplement l'offre de formation indépendamment de l'établissement responsable de ces formations. Nous avons pu confronter nos connaissances théoriques à une réalité pratique. Nous avons appris et goûté le développement d'un vrai projet informatique.

Le plus important pour nous, était d'avoir pu contribuer dans un grand projet qui est la création d'un Environnement Numérique de Travail (ENT) à l'université Abou Bekr Belkaid Tlemcen, tout en sachant que notre travail n'est qu'un module parmi tant d'autre. L'ENT est loin d'être fini et nous espérons contribuer davantage dans le future avec le développement d'autres modules mais surtout par sa généralisation à travers toutes les universités algériennes et pourquoi pas maghrébines également.

Bibliography

- 1: l'office des publications universitaires 1, guide pratique de mise en œuvre et de suivi du lmd, Place centrale- Ben Aknoun – ALGER, ,
- 2: Academie Languedoc-Roussillon, Présentation générale de ENT, ,<http://www.environnementnumeriquedetravail.fr/uPortal/f/u2711s4/normal/render.uP>
- 3: wikipedia, Java EE, 2015,http://fr.wikipedia.org/wiki/Java_EE
- 4: wikipedia, JavaServer Pages, ,http://fr.wikipedia.org/wiki/JavaServer_Pages
- 5: Sql.sh, Apprendre le SQL, ,<http://sql.sh/>
- 6: W3C, HTML5, ,<http://www.w3.org/TR/html5/>
- 7: futura-sciences.com, CSS, ,<http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-css-4050/>
- 8: Johann Pardanaud, Introduction au JavaScript, ,<http://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript/introduction-au-javascript>
- 9: java.developpez.com, RAD et EDI Java, ,<http://java.developpez.com/telecharger/index/categorie/336/RAD-et-EDI-Java>
- 10: mysql.fr, MySQL WorkBench, ,<https://www.mysql.fr/products/workbench/>
- 11: .umlet.com, UMLet 13.2Free UML Tool for Fast UML Diagrams, ,<http://www.umlet.com/>
- 12: Médéric Munier, Créez votre application web avec Java EE , ,<http://openclassrooms.com/courses/creez-votre-application-web-avec-java-ee>

Résumer

Ce document a été présenté comme Mémoire de fin d'études pour l'obtention du diplôme de Licence en Informatique sous le thème "la Mise en place d'un catalogue de formation détaillé de la faculté de sciences".

Dans le premier chapitre on a présenté le système LMD, ensuite dans le deuxième on a cité les langages et les technologies utilisés lors du développement de notre projet, et dans le troisième chapitre on a passé à la conception de notre modèle de gestion de formation, et à la fin nous avons terminé par la présentation de notre solution et les différentes étapes de sa réalisation.

ملخص عام

تم تقديم هذا البحث في إطار مشروع نهاية الدراسة لنيل شهادة الليسانس تحت عنوان تطوير دليل رقمي لفرص التكوين للجامعة وقد جاء فيه عرض سريع لمبادئ نظام التعلم العلي الجديد (لسانس - ماستير - دوكتوراه) وورد هذا في المبحث الأول, أما المبحث الثاني فتم التطرق فيه إلى مختلف لغات البرمجة و الأدوات التقنية المستخدمة في عملية التطوير, وبغرض تقريب القارئ من تفاصيل التقنية للموقع المنجز جاء المبحث الثالث بمجموعة من المخططات البيانية والتوضيحية, هذا وقد خصص المبحث الأخير لسرد مراحل انجاز المشروع وتوضيح طريقة استعماله

summary

This document has been presented as a memo for obtaining a licence degree in computer sciences and the theme is "catalogue of formation offered by faculty of sciences".

In the first chapter we have presented what the new system called "LMD" stand for, then in the second chapter we talked about the languages and the technologies choices.

In the third chapter we moved to design and conception phase and finally we presented our solution which is a web based application and the steps to achieve that application.