



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE



Mémoire

de fin d'études

En vue de l'obtention du diplôme de Master en informatique

Option : Ingénieries des Systèmes d'Information (ISI)

Thème :

**Développement d'un Système d'Aide à
l'Apprentissage de Compétences
Disciplinaires**

Proposé et dirigé par :

Mme BOUARAB.F

Réalisé par :

Mr YAHIA Mohamed
Mr BECHIRI Marzouk

Promotion : 2013/2014

Remerciements

Nous remercions tout d'abord Dieu qui nous a donné la foi et le courage pour accomplir ce projet.

Nous tenons aussi à exprimer notre reconnaissance et profonde gratitude à notre promotrice Mme BOUARAB.F pour nous avoir encadré durant cette année, pour sa forte présence et sa disponibilité, pour son exigence scientifique et ses précieuses orientations méthodologiques, pour son encouragement et sa patience.

Que les membres du jury trouvent ici nos plus vifs remerciements pour avoir accepté d'honorer par leur jugement notre travail. Aussi, nous adressons nos remerciements à tous nos enseignants de *l'UMMTO* pour nous avoir appris le goût de l'effort et du travail.

Un grand merci aussi à toute personne qui de près ou de loin a contribué à ce que cet humble travail voit le jour.

Dédicaces

Je dédie cet humble travail:

À mes grands parents

À mes parents adorés

À mes sœurs et mon frère

À mon oncle et sa famille

*Et à tous mes amis(e) et tous ceux qui me sont chers au
monde.*

Marzouk.

Dédicaces

Je dédie ce modeste travail :

À mes très chers parents ;

À mon frère ;

À mes sœurs et leurs maris Ali, Mourad et Samir ;

*Et à tous mes amis (e) et tous ceux qui me sont chers au
monde.*

Mohamed

Sommaire

Introduction générale	1
Chapitre I : Fondements théoriques	
Introduction	3
I.1. Les principales théoriques de l'apprentissage	3
I.1.1. Le behaviorisme (comportementalisme) et l'approche par objectifs	4
I.1.1.1. Développement du behaviorisme	4
I.1.1.2. Applications pédagogiques du behaviorisme.....	4
I.1.2. Le cognitivisme	6
I.1.2.1. Développement du cognitivisme	6
I.1.2.2. Applications pédagogiques du cognitivisme.....	7
I.1.3. Le constructivisme et le socioconstructivisme.....	8
I.1.3.1. Développement du constructivisme et du socioconstructivisme	8
I.1.3.2. Applications pédagogiques du constructivisme	8
I.2. Pourquoi changé de paradigme	9
I.2.1. Les limites du behaviorisme et les dérives de la pédagogie par objectifs ..	9
I.3. l'approche par compétence et les systèmes hypermédia adaptatif éducatif	10
I.3.1. L'approche par compétence	10
I.3. 2. La pédagogie par situation problème	11
I.3. 3. Hypermédia Adaptatif	11
I.3. 3. 1. Principe et définition	11

I.3. 3. 2. Eléments d'adaptation.....	12
I.3. 4. Classification des hypermédias adaptatifs	12
I.3. 4. 1. Les hypermédias adaptatifs éducatifs	12
I.3. 4. 2. Hypermédias adaptatifs dynamiques	13
I.3.5. Architecture d'un hypermédia adaptatif éducatif avec l'approche par compétence	14
I.3.5. 1. Description des composantes.....	14
I.3.5. 1.1. Le modèle du domaine	14
I.3.5. 1.2. Le modèle apprenant	17
I.3.5. 1.3. Générateur	17
Conclusion.....	17

Chapitre II : Analyse et Conception

II.1. Analyse	19
II.1.1. Identification des acteurs	19
II.1.2. Architecture globale du système	19
II.1.2.1. Spécification de la fonction d'évaluation.....	21
II.1.2.2. Spécification de la Génération de plan de guide pour le filtrage des compétences.....	22
II.1.2.3. Spécification de la Génération de plan de guide pour le filtrage des situations de problèmes.....	23
II.1.3 Analyse avec UML	24
II.1.3.1. Modèle des cas d'utilisation	24
II.1.3.1.1. Spécification des taches	24
II.1.3.1.2. Spécification des scénarios.....	25

II.1.3.1.3. Identification et description des cas d'utilisation	27
II.1.3.1.4. Ajout des relations entre cas d'utilisation	29
II.1.3.1.5. Finalisation des diagrammes de cas d'utilisation	30
II.2. Conception avec UML.....	33
II.3.1. Elaboration des diagrammes de séquence	33
II.3.2. Diagrammes d'activité	36
II .3.3. Diagramme de classe	38
II .3.4. Le modèle physique des données	41
Conclusion.....	46

Chapitre III : Réalisation

Introduction	47
III.1. Outils utilisés	47
III.1.1. Langages à balises	47
III.1.2. Langages de script	49
III.1.3. Serveurs	50
III.2. Environnement de développement.....	52
III.2.1. langage de programmation java	52
III.2.2. Architecture MVC	53
III.2.3. Avantages de l'architecture MVC	56
III.2.4. Différence avec l'architecture trois tiers	57
III.3. Les outils de développement.....	57
III.3.1. IDE NetBeans.....	57

III.3.2. Le SGBD (MYSQL 5)	59
III.4. Présentation de quelques interfaces	60
III.4.1. Page d'accueil	60
III.4.2. Page d'authentification et d'inscription apprenant	61
III.4.3. Espace Apprenant	62
III.4.4. Page d'authentification Expert du domaine	64
III.4.5. Espace expert du domaine	65
Conclusion	72
Conclusion Générale	73

✓ Liste des figures et tableaux

Figure I.1 : Architecture d'un hypermédia adaptatif avec l'approche par compétence.....	14
Figure I.2 : Lien de composition entre compétences	16
Figure I.3 : Relation de pré requis	16
Figure II.1. Architecture globale du système.....	20
Figure II.2. Spécification de la fonction d'évaluation.....	21
Figure II.3 : Spécification de la Génération de plan de guide pour le filtrage des compétences	22
Figure II.4 : Spécification de la Génération de plan de guide pour le filtrage des situations de problèmes	23
Figure II.5 : spécification des taches	25
Figure II.6 : spécification des scénarios	27
Figure II.7 : Diagramme préliminaire des cas d'utilisation	29
Figure II.8: Diagramme des cas d'utilisation «Expert de domaine»	31
Figure II.9 : Diagramme de cas d'utilisation « Apprenant»	32
Figure II.10: Diagramme de séquence du cas d'utilisation « s'authentifier»	34
Figure II.11 : Diagramme de séquence du cas d'utilisation « résoudre une situation de problème ».	35
Figure II.12 : Diagramme d'activité pour le cas d'utilisation «s'authentifier»	36
Figure II.13: Diagramme d'activité pour le cas d'utilisation «Répondre à une situation de problème»	37
Figure II.14 : Diagramme de classe générale de cas d'utilisation «Authentification»	38
Figure II.15: Diagramme de classe général de cas d'utilisation «Résoudre une situation de problème»	39
Figure II.16 : Diagramme de classe global	40
Figure III.1 : exemple de fichier HTML	47
Figure III.2 : exemple du code Style CSS	48
Figure III.3 : exemple de script jQuery	50

Figure III.4 : interface "patern MVC"	56
Figure III.5 : interface "Netbeans 8.0"	58
Figure III.6 : interface "MySQL Workbench"	59
Figure III.7 : interface d'accueil	60
Figure III.8 : Page d'authentification et d'inscription apprenant	61
Figure III.9 : Espace Apprenant	62
Figure III.10:Interface résoudre une situation de problème	63
Figure III.11:interface d'authentification Expert du domaine.....	64
Figure III.12: interface mise à jour compétence générale.....	65
Figure III.13: interface mise à jour compétence composée.....	66
Figure III.14: interface mise à jour compétence élémentaire.....	67
Figure III.15: interface ajout des compétences générales sans pré-requis	68
Figure III.16 : Espace Apprenant affichage des compétences sans pré-requis	69
Figure III.17: interface modification une compétence générale avec pré-requis.....	70
Figure III.18 : Espace Apprenant non affichage des compétences qui ont des pré-requis.....	71
Figure III.19: interface mise à jour situation de problème	72

Introduction Générale

L'approche par compétence est une démarche dans laquelle sont engagés les systèmes adaptatifs éducatifs. Elle consiste à définir les compétences dont chaque apprenant a besoin pour passer à l'étape suivante de son parcours d'apprentissage, pour accéder à une qualification et pour un apprentissage plus concret, plus actif et plus durable.

L'objectif de notre travail est d'intégrer cette approche dans un système d'aide à l'apprentissage, cette approche va permettre d'assurer que tous les apprenants développent les mêmes compétences essentielles, et de développer des niveaux de compétences élargis selon les capacités individuelles des apprenants.

Pour réaliser ce travail, nous avons organisé notre mémoire de la manière suivante :

- **Le premier chapitre** présente les principales théoriques de l'apprentissage, sous les deux approches pédagogiques (objectif et compétence), et l'intégration de l'approche par compétence (PPC) dans les systèmes hypermédias adaptatifs éducatifs.
- **Le deuxième Chapitre** est consacré à la démarche méthodologique à suivre afin de planifier et concevoir l'application d'une façon rigoureuse tout en mettant en évidence tous les objectifs fixés en amont du projet et veiller à son élaboration.
- **Le troisième chapitre** : Dans ce chapitre, nous présentons les différentes étapes de l'implémentation de notre application. nous avons consacré une première partie au choix de langage de développement, l'environnement et les outils d'implémentation de notre application. Ensuite dans la deuxième partie nous avons mis en œuvre des différentes interfaces de notre site (application), et leurs fonctionnalités essentielles.

Enfin, nous terminons par une conclusion générale dans laquelle sont présentées les perspectives relatives à ce travail.

Chapitre I :
Fondements
théoriques

I. Fondements théoriques

Introduction

Dans la philosophie de l'apprentissage plusieurs théories ont vu le jour, leur but est de comprendre et de modéliser les processus d'apprentissage. La théorie behavioriste (comportementaliste), qui à l'origine de la pédagogie par objectifs (PPO), est la première à avoir été développée.

Ainsi, des nouvelles théories, la théorie de cognitiviste, la théorie constructiviste et la théorie socioconstructiviste tentant l'explication du processus d'acquisition des connaissances chez les êtres humains sont nées. La théorie cognitiviste s'intéresse à la façon dont l'individu apprend. Pour les constructivistes, l'apprentissage est défini comme un processus actif de construction des connaissances, l'apprenant construit ses propres connaissances en interagissant avec son environnement. Le socioconstructivisme met l'accent sur le rôle socio-historique et les interactions sociales dans la construction des connaissances.

I.1. Les principales théoriques de l'apprentissage [APP,2011]

Le behaviorisme, le cognitivisme, le constructivisme et le socioconstructivisme sont les grands paradigmes qui ont marqué le monde de l'éducation depuis le milieu du vingtième siècle. Un paradigme étant « un système de croyances fondamentales ou une vision du monde » [Guilbert, 2000], ce sont des cadres de références qui standardisent la construction des savoirs. « Ils sont partagés par les chercheurs qui travaillent au sein d'une même communauté scientifique » [Savoie-Zajc et Karsenti, 2000]. Il importe de se référer à ces paradigmes parce qu'ils sont déterminants pour les théories de l'apprentissage, l'apprentissage étant vu différemment selon qu'on s'inscrit dans l'un ou l'autre des paradigmes.

I.1.1. Le behaviorisme (comportementalisme) et l'approche par objectifs

I.1.1.1. Développement du behaviorisme

Pour les behavioristes, l'apprentissage est défini comme une modification durable des comportements, un comportement étant l'ensemble des réactions objectivement observables d'un organisme qui réagit à un stimulus. L'objectivisme qui stipule qu'il existe une réalité externe, objective, séparée de la conscience est une croyance fondamentale des behavioristes. Par ailleurs, pour les behavioristes les comportements des individus sont déterminés par des conditions de l'environnement. Ces conditions façonnent les comportements, ce qui sera déterminant dans le choix de stratégies pédagogiques présentées dans les paragraphes suivants.

I.1.1.2. Applications pédagogiques du behaviorisme

Les applications de cette théorie en éducation sont très nombreuses, on retrouve particulièrement l'enseignement programmé, la pédagogie de la maîtrise et surtout l'approche par objectifs, appelée également la pédagogie par objectifs.

Le courant de la pédagogie par objectifs a pris son essor grâce aux travaux de **Bloom** dont la taxonomie est renommée. En 1956, [**Bloom**] établit la taxonomie des objectifs pédagogiques du domaine cognitif, psychomoteur et affectif qui sera largement utilisée en enseignement programmé puis en technologie de l'éducation. [**Mager**] un autre chercheur va, en 1962, marquer à son tour le courant de la pédagogie par objectifs en proposant une formulation très précise des objectifs pédagogiques. Pour ce dernier, un objectif pédagogique doit contenir un verbe d'action qui décrit la performance attendue de l'étudiant, une description des conditions de réalisations et les critères de performance. De nos jours, les objectifs pédagogiques de [**Bloom** et **Mager**] sont encore largement utilisés dans la conception de programmes de formation, d'activités de formation et de cours.

De manière générale, toute conception de système de formation ou tout design pédagogique demande une planification qui est un processus consistant à atteindre des

objectifs de formation en tenant compte de plusieurs aspects : les besoins de l'apprenant, le contexte dans lequel l'apprentissage s'effectuera, la population visée par l'apprentissage, l'analyse de contenus, le choix des moyens et des médias et, enfin, l'évaluation. Comme on le verra plus loin, ce processus varie en fonction du paradigme dans lequel on s'inscrit. Qu'en est-il dans une perspective behavioriste? La description qui suit est inspirée du travail de [Basque, 1999] qui a analysé l'influence des théories de l'apprentissage sur le design pédagogique.

- ❖ Le processus de design pédagogique ne tient pas compte des besoins d'apprentissage de l'apprenant. L'apprenant n'est pas impliqué dans le processus de définition de ses propres besoins. Ce qui importe c'est la performance visée par les responsables de formation. À cet effet, l'écart entre la performance actuelle de l'apprenant et la performance visée par les responsables de formation constitue le besoin de formation.
- ❖ **L'analyse du contexte** (physique, administratif, etc.) permet de contrôler les possibles effets négatifs des conditionnements sur les attitudes et le niveau de participation des étudiants.
- ❖ **L'analyse de tâches** consiste à décortiquer chaque tâche en niveau hiérarchique (sous tâches, sous-sous-tâches, etc.), le but de cet exercice étant de déterminer les objectifs qui correspondent à chaque niveau de tâches, de les traduire en comportements observables et de les évaluer. Cette décomposition permet également d'identifier les types d'apprentissage et de développer les concepts sous-jacents à l'acquisition de ces comportements. L'enseignement est ainsi divisé en séquences d'enseignement.
- ❖ **Les objectifs d'apprentissage** sont présélectionnés par le concepteur pédagogique. Il indique très exactement les comportements observables, c'est-à-dire ce que l'apprenant sera capable de faire à la fin de son enseignement. Généralement, les objectifs sont formulés selon les critères de formulation de [Mager, 1962] qui stipule qu'un objectif pédagogique doit contenir un verbe d'action qui décrit la performance attendue de l'étudiant, une description des conditions de réalisation de cette performance et un critère de performance.

- ❖ L'enseignant s'engage dans une classification des objectifs à atteindre, en allant du plus simple au plus complexe. Il élabore un plan d'apprentissage comportant autant d'objectifs que de comportements à acquérir.
- ❖ L'**évaluation** des apprentissages porte sur les comportements observables, elle peut être de type formatif, sommatif ou critérié. L'évaluation critériée compare dans ce cas la performance de l'apprenant au niveau d'acquisition des objectifs. L'examen objectif comportant un choix de réponses est le plus utilisé.
- ❖ **Les stratégies pédagogiques** visent à amener les étudiants à manifester le comportement ciblé en commettant le moins d'erreurs possible. Ces stratégies mettent l'accent sur la présentation de contenu, les exercices répétitifs et la révision systématique.
- ❖ **Les médias** sont considérés comme des stimuli auditifs, visuels, tactiles, etc., permettant d'atteindre le comportement désiré [**Marquis et Sauvé, 1995**]. Les behavioristes privilégient particulièrement les moyens audiovisuels (diaporama, radio, télévision, film, machine à enseigner) qui permettent de présenter des contenus ou des démonstrations.

Qu'en est-il de nos jours des applications pédagogiques du behaviorisme? On constate que le behaviorisme est toujours utilisé, plusieurs activités de formation continue, telles que les colloques et les symposiums, sont souvent élaborées suivant le modèle behavioriste.

I.1.2. Le cognitivisme

I.1.2.1. Développement du cognitivisme

Les découvertes réalisées par les sciences cognitives au cours des trois dernières décennies fournissent de précieux renseignements sur la façon dont l'être humain traite l'information provenant de l'environnement [**Bissonnette, Gauthier et Richard, 2005**]. Les sciences cognitives s'intéressent particulièrement aux opérations mentales telles que le raisonnement, la mémorisation, la résolution de problème et le transfert des connaissances. La motivation est un aspect très important du processus d'apprentissage chez les cognitivistes.

Pour les cognitivistes, l'apprentissage est vu comme un changement dans les structures mentales ou représentations internes des individus. C'est un processus actif de traitement de l'information et de résolution de problèmes. Pour [Basque, 1999 et Brien, 1997], la vision de l'éducation qui découle de ce paradigme met l'accent sur l'engagement actif des apprenants durant l'apprentissage afin qu'il puisse traiter les informations en profondeur. Dans cette perspective pour créer des situations favorables et motivantes à l'apprentissage, il faut :

- ❖ tenir compte des particularités individuelles étant donné que chaque individu traite l'information selon sa propre représentation;
- ❖ favoriser des activités faisant appel à la mémoire à long terme;
- ❖ mettre en place des stratégies qui font appel à la résolution de problèmes et qui développent le processus métacognitif des apprenants

I.1.2.2. Applications pédagogiques du cognitivisme

Les premières applications du cognitivisme en éducation ont été réalisées au milieu des années 1970 par [Robert Gagné]. Il a mis au point un modèle qui décrit la façon dont l'esprit humain traite l'information en situation d'apprentissage et comment se fait la mémorisation. À partir des années 80, on commence à concevoir des modèles de système de formation (design pédagogique) d'influence cognitiviste. À titre d'exemple, au Québec, [Brien, 1997] préconise un modèle de design qui est un processus allant de l'analyse de besoins jusqu'à l'implantation, en passant par la formulation des objectifs ou compétences, la structuration du contenu, le choix des méthodes d'enseignement et la mise à l'essai du système. Ce modèle constitue, à quelques différences près, l'ossature des modèles de design d'inspiration cognitiviste.

I.1.3. Le constructivisme et le socioconstructivisme

I.1.3.1. Développement du constructivisme et du socioconstructivisme

Pour les constructivistes, l'apprentissage est défini comme un processus actif de construction des connaissances et non pas comme un processus d'accumulation et d'acquisition de connaissances. L'apprenant est un être proactif qui construit ses propres connaissances et qui interagit avec son environnement. Il construit ses connaissances au cours de ses propres expériences et, de ce fait, la connaissance est vue comme le résultat des activités d'un être actif.

Par ailleurs, pour les constructivistes il n'existe pas qu'une seule « vision du monde », car chaque individu interprète la réalité selon ses connaissances, ses expériences, ses compétences, selon son cadre de référence. Pour les constructivistes, les réalités sont multiples.

Pour les socioconstructivistes, la conscience et la pensée ne constituent pas des caractéristiques purement internes étant donné que ces dernières s'élaborent à partir d'activités externes réalisées dans un environnement social déterminé. Il insiste sur l'importance des interactions avec autrui afin de prendre conscience de ses propres actions et de son processus de pensée et sur le rôle de la culture qui est déterminant dans la formation de la pensée.

[Piaget ,1896-1980] et [Vygotsky, 1896-1934] sont reconnus respectivement comme les principaux pionniers du constructivisme et du socioconstructivisme. Tous deux, critiques à l'égard du behaviorisme et de l'approche réductionniste (fondée sur le postulat selon lequel tout système complexe peut être expliqué en la réduisant en ses éléments simples), ont influencé, de manière importante, le domaine de l'éducation et de la recherche en pédagogie grâce à leurs divers travaux de recherche.

I.1.3.2. Applications pédagogiques du constructivisme

Les perspectives constructivistes ont acquis une grande popularité en éducation et sont fréquemment évoquées à titre de fondements de divers courants pédagogiques. L'idée principale qu'on associe à cette nouvelle vision de l'enseignement et de l'apprentissage est celle de l'apprenant maître de son apprentissage, qui construit son

propre savoir au contact de l'environnement et en interaction avec les autres. De ce fait, le rôle de l'enseignant change radicalement puisqu'il n'est plus le seul et unique détenteur des connaissances comme il l'était dans l'enseignement traditionnel.

I.2. Pourquoi changer de paradigme

I.2.1. Les limites du behaviorisme et les dérives de la pédagogie par objectifs

De nos jours, les critiques du behaviorisme en général et de l'approche par objectifs en particulier, ainsi que de leurs retombées sur les programmes de formation, sur les compétences développées par les apprenants et sur leur attitude face à l'apprentissage, sont nombreuses. Bien que cette théorie soit à l'origine de nombreux progrès en éducation, elle semble désormais moins pertinente pour développer des systèmes de formation. Les principales critiques relevées dans les écrits sont décrites brièvement dans les lignes qui suivent.

- ❖ Dans une approche par objectifs, l'obligation de décrire de manière exhaustive tous les apprentissages visés par l'apprenant et d'associer un objectif à chaque tâche conduit à une prolifération d'objectifs pédagogiques difficiles à gérer par l'enseignant.
- ❖ Cette décomposition des savoirs conduit au morcellement des connaissances et à l'atomisation des compétences. Ce découpage en petites unités abordées selon une suite logique et progressive ne permet pas à l'apprenant d'avoir une vision d'ensemble de sa formation et, ainsi, de faire des liens entre les connaissances abordées, ce qui ne favorise pas l'intégration des savoirs.
- ❖ L'approche par objectifs se concentre généralement sur des habiletés de bas niveau intellectuel telles que la mémorisation, la définition et l'illustration de concepts, l'application ou l'exécution.
- ❖ L'approche par objectifs est plutôt centrée sur le contenu et sur l'accumulation des connaissances. En conséquence, l'évaluation s'intéresse généralement aux objectifs d'ordre cognitif reliés à la discipline. Pour [Legendre, 2001], ces connaissances sont souvent des connaissances inertes étant donné l'incapacité de l'apprenant à les réutiliser de façon fonctionnelle et judicieuse dans d'autres situations. Pour [Lebrun, 2007], les

connaissances sont inertes parce qu'elles ne sont pas contextualisées et de ce fait difficilement mobilisables. En effet, toujours selon [Legendre, 2001], de nombreuses recherches réalisées dans un contexte scolaire et dans divers domaines du savoir ont mis en évidence le peu de réinvestissement et de transfert des acquis.

- ❖ L'évaluation au détriment de l'apprentissage est également une critique relevée par de nombreux chercheurs [Scallon, 2005 et Tardif, 2006]. La manifestation observable étant une des caractéristiques essentielles du behaviorisme, l'accent est mesurée des évaluations partielles qui sont souvent elles-mêmes associées à une tranche de connaissances bien particulières.

I.3. L'approche par compétence et les systèmes hypermédia adaptatif éducatif

I.3.1. L'approche par compétence [Berk & boua]

L'approche par compétences (APC) imposée dans les programmes scolaires de nombreux pays, permet de passer d'une pédagogie centrée sur les savoirs (savoirs scolaires), à une pédagogie centrée sur des activités dans lesquelles ces savoirs s'incarnent. Ces savoirs sont en construction dynamique et en permanence recombinaison par l'élève en vue d'un apprentissage de savoir-faire.

Deux courants de pensée définissent la notion de compétence. Dans le premier courant, les auteurs réduisent la formulation des compétences à des habilités, portant strictement sur des contenus notionnels reliés à des savoirs disciplinaires traditionnels. Dans le second courant, la notion de compétences réfère aux résultats des actions de personnes traitant des situations problèmes.

I.3. 2. La pédagogie par situation problème

Une situation problème est une tâche concrète adaptée aux élèves, imaginée par un enseignant comme un espace de réflexion et d'analyse autour d'une question à résoudre. Afin d'accomplir cette tâche, l'enseignant est supposé choisir un ensemble de ressources (savoirs, savoir-faire et savoir-être) et de contraintes (matérielles et humaines) en fonction des compétences souhaitées, à développer chez les élèves.

Cependant pour mieux piloter l'apprentissage des apprenants, ces derniers doivent être observés et leurs compétences évaluées. L'observation porte sur les actions pédagogiques qu'ils effectuent durant leur apprentissage.

I.3. 3. Hypermédia Adaptatif

I.3. 3. 1. Principe et définition

Des recherches ont alors essayé de minimiser l'aspect négatif des hypermédias en créant des hypermédias adaptatifs. L'objectif de ces systèmes est d'adapter la présentation de la connaissance et d'aider l'apprenant à se diriger dans l'hyperespace. De ce fait, dans un hypermédia nous devons pouvoir modifier aussi bien le contenu des pages que les liens entre les différentes pages [Bru, 96]. Mais c'est surtout sur l'adaptation des liens que le plus grand nombre de techniques ont été développées [Bru, 96]. On trouve par exemple des techniques de guidage direct, des techniques d'ordonnement des liens, des techniques de masquage des liens ou bien encore des techniques d'annotation des liens.

L'architecture des hypermédias adaptatifs, comme pour beaucoup de systèmes d'enseignement assisté par ordinateur, s'appuie principalement sur deux modèles : le modèle du domaine et le modèle de l'élève. Les différents types d'hypermédias adaptatifs se caractérisent par la relation qu'ils entretiennent entre le modèle du domaine et les média utilisés pour présenter les concepts à l'apprenant. Les systèmes ont successivement employé différentes techniques (par exemple la méthode des pages d'index ou la méthode d'indexation fragmentée). Mais la technique la plus évoluée calque la structure de l'hyperespace sur la structure du modèle du domaine [Vas, 97]. Ainsi chaque

concept est relié à une ou plusieurs pages physiques et ces relations sont représentées par des liens hypertextes.

I.3. 3. 2. Eléments d'adaptation

a. La présentation adaptative

Le principe de la présentation adaptative est d'adapter le contenu des pages de l'hypermédia à un utilisateur particulier en fonction de ces connaissances, ces objectifs et ces préférences. En d'autres termes, un utilisateur considéré comme qualifié peut être alimenté avec des informations plus approfondies quant à un novice se contente uniquement des informations générales. De ce fait, les utilisateurs qui accèdent à une même page, mais ayant des profils différents, doivent visualiser en fait des pages différentes.

b. La navigation adaptative

L'idée de la navigation adaptative est de guider l'utilisateur dans l'hyperespace en adaptant la façon de présentation des liens aux objectifs visés par l'utilisateur, à son niveau de connaissance et à d'autres caractéristiques tirées de son modèle.

I.3. 4. Classification des hypermédias adaptatifs

Les systèmes hypermédias adaptatifs ont été très utiles dans plusieurs domaines : l'extraction de la connaissance, l'aide en ligne, l'éducation, etc. En effet, une analyse des systèmes hypermédias a permis de les classer en six domaines d'application.

I.3. 4. 1. Les hypermédias adaptatifs éducatifs

L'idée d'intégrer l'hypermédia à l'enseignement n'est pas récente, elle remonte aux années 60 (machine Autotutor Mark II) où l'accent a été mis sur l'individualisation de l'enseignement et le travail autonome de l'apprenant en utilisant les différents médias (TV éducative, Radio, Film, Vidéo, etc.). [Catano, 1979], a été le premier à utiliser un système hypertexte (FRESS) pour enseigner la poésie. Depuis, les recherches n'ont pas cessé pour intégrer des ressources multimédias dans l'éducation [Duf, 98].

Avantages

Les avantages de l'utilisation des hypermédias dans l'enseignement sont les suivants

- La rapidité et la facilité avec lesquelles un apprenant peut accéder à l'information.
- La comparaison, la confrontation et l'analyse d'éléments d'information rapprochés. Ce qui donne ainsi plusieurs points de vue sur un même objet (connaissances, théoriques, illustrations, explications, études de cas, simulations, etc.).
- L'entraînement à l'évaluation, à l'assimilation par le rajout des annotations diverses, etc.

L'apprenant peut acquérir par « imprégnation » non seulement la connaissance

consciemment recherchée, mais une part sans doute non négligeable de tout ce qu'il a fortuitement rencontré. «... Fournir à l'étudiant un environnement stimulant, en lui laissant le choix de son objet d'étude, lui offrir une bonne variété de matériel intéressant ... il se trouvera motivé, intéressé à en faire un peu plus dans le cadre de l'enseignement traditionnel » [Nel, 70].

I.3. 4. 2. Hypermédias adaptatifs dynamiques

L'objectif de ces systèmes est d'adapter la présentation de la connaissance et aider l'apprenant à naviguer à travers le graphe composé par l'ensemble des pages et des liens hypermédia. Selon **Brusilovsky**, ils sont de deux sortes :

- Les systèmes qui adaptent le contenu présenté à l'apprenant
- Les systèmes qui adaptent les liens présentés à l'apprenant

I.3.5. Architecture d'un hypermédia adaptatif éducatif avec l'approche par compétence

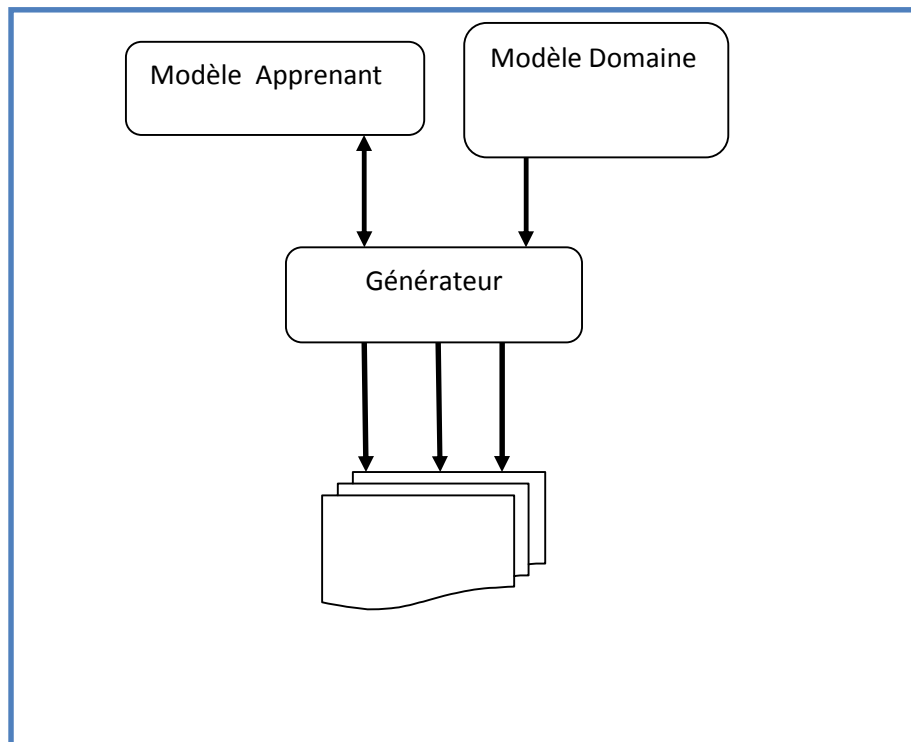


Figure I.1 : Architecture d'un hypermédia adaptatif avec l'approche par compétence

I.3.5. 1. Description des composantes

I.3.5. 1.1. Le modèle du domaine

a. Les compétences

La définition adoptée pour la notion de compétence est celle de Tardif [Tardif, 2006] qui considère une compétence comme « un savoir-agir complexe qui prend appui sur la mobilisation et l'utilisation efficace d'une variété de ressources internes et externes à l'intérieur de situations problèmes ». On distingue des compétences générales, des compétences composées et des compétences élémentaires:

- **Une compétence élémentaire (CE):** est la composante la plus élémentaire dans un domaine.

- **Une compétence composée(CC)** : est une compétence composée de plusieurs compétences élémentaires.
- **Une compétence générale(CG)** : est composée d'un ensemble de compétences composées et élémentaires.

Dans chaque compétence, on associe un ensemble des situations problèmes (SP), dans lesquelles les compétences vont se développer.

❖ **Situation de problème** : est une tâche à réaliser dans une discipline donnée (dans notre cas : base de donnée), adaptée aux profils d'apprenants (chacun sa progression, ses acquis antérieurs, ses expériences, etc.). Chaque situation de problème d'un degré de complexité et d'une durée donnée, est accompagnée de :

- ✓ **Une consigne** : Est un guide sous forme d'énoncé qui indique le but à atteindre ainsi que la démarche pouvant l'atteindre.
- ✓ **Un obstacle** : Les obstacles sont liés aux « manques » des apprenants : de connaissances, de savoir-faire ou absence de comportement et d'attitudes adéquates, un apprenant peut s'affronter à des obstacles spécifiques (propres à la discipline) ou généraux (liés à plusieurs disciplines).
- ✓ **Un critère** : Est une norme pour évaluer les actions des apprenants.

b. Les relations

En plus des relations de composition (est composée de) entre les compétences, on trouve des relations de type « pré requis de » entre ces compétences.

- ❖ La relation « **Est composé de** » existe entre les compétences comme le montre la figure I.3.

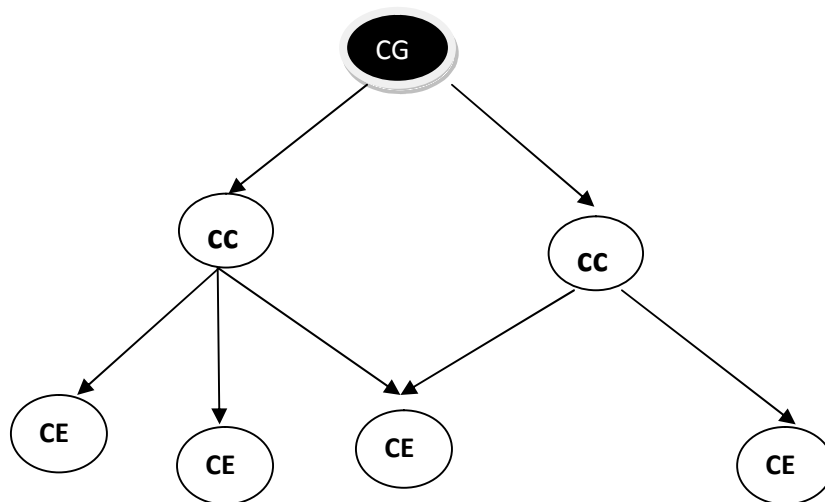


Figure I.2 : Lien de composition entre compétences

- ❖ La relation « **pré requis de** » : permet de spécifier quelle compétence qu'on doit afficher avant l'autre.

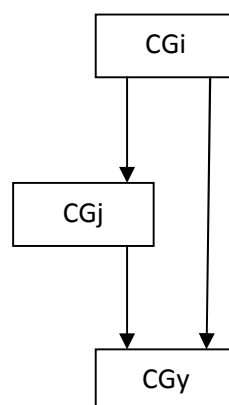


Figure I.3 : Relation de pré requis

Dans la figure ci-dessus, CGi, CGj et CGy sont des compétences générales. Sur cette figure, CGi est un pré requis direct de CGj et CGj est un pré requis direct de CGy donc CGi est un pré requis obligatoire de CGy et CGj est un pré requis optionnel CGy. C.-à-d tant que l'apprenant n'aura pas acquis CGi, il ne pourra pas consulter CGj et CGy.

I.3.5. 1.2. Le modèle apprenant : est la composante qui permet de représenter aussi fidèlement que possible les caractéristiques de l'apprenant (ses expériences, ses acquis antérieurs, etc.). Il sert de déclencheur de générateur.

I.3.5. 1.3. Générateur : son rôle est de générer des compétences aux quelles l'apprenant doit s'évaluer, prenons en compte de son profile (l'état de son avancement) ainsi que les relations entre les compétences (pré requis).

Conclusion

Dans ce chapitre on a présenté les différentes théories d'apprentissage sous les deux approches, (l'approche par objectif et par compétence), en signalant les dérives de l'approche par objectif. Nous avons étudié l'approche par compétence dans les systèmes hypermédias adaptatifs éducatifs.

Le chapitre suivant, fera l'objet de la conception de notre application.

Chapitre II :
Analyse et
Conception

Introduction :

Dans le but d'une meilleure organisation et une bonne maîtrise du travail, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou démarche bien définie.

Pour notre projet, nous allons suivre un processus de développement d'application web, en utilisant le langage de modélisation très répandu UML (Unified Modeling Language) et son extension pour le Web, qui permet de bien représenter l'aspect statique et dynamique d'une application par une série de diagrammes qu'il offre.

Les diagrammes d'UML peuvent être utilisés librement à tous les niveaux d'un processus d'élaboration d'un produit logiciel. Cependant, la pratique montre bien que certains diagrammes s'apprêtent bien mieux à certaines phases qu'aux d'autres.

II.1. Analyse

Cette activité commence par la mise en évidence des différents acteurs intervenants dans le système cible ainsi que leurs besoins.

II.1.1. Identification des acteurs

Les acteurs sont des entités qui représentent des rôles à travers une certaine utilisation (cas). Notre application fait intervenir deux acteurs :

- **Apprenant:** Est toute personne concernée par l'apprentissage par compétence au sein du site, après avoir finalisé son inscription.

- **Expert de domaine :** C'est les professeurs ou tuteurs. Parmi les fonctionnalités offertes, il y a l'ajout, modification et la suppression des différentes compétences, situations de problèmes et consignes.

II.1.2. Architecture globale du système

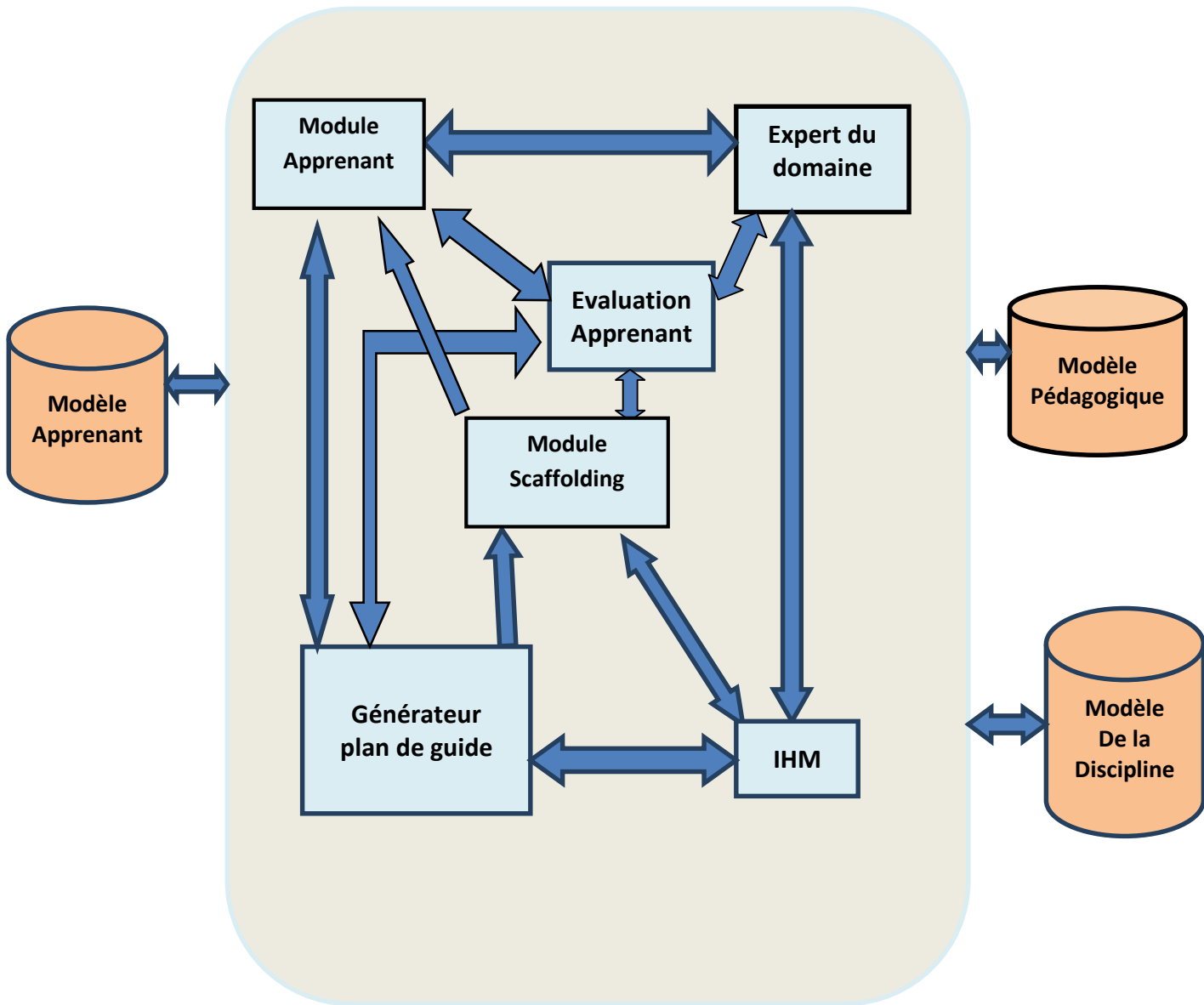


Figure II.1 : Architecture global du système

II.1.2.1. Spécification de la fonction d'évaluation

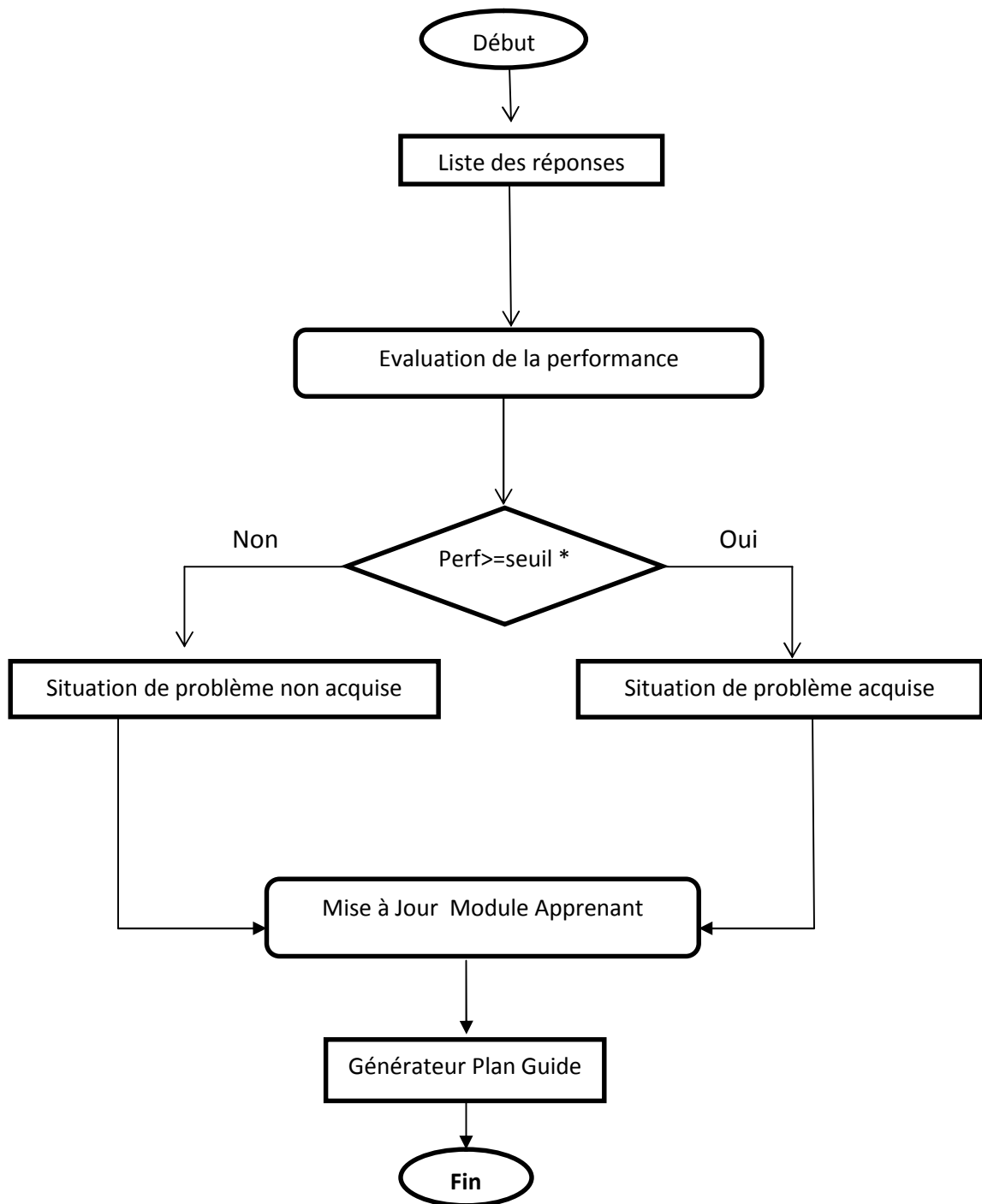


Figure II.2 : Spécification de la fonction d'évaluation

Seuil* : c'est le seuil de la performance de la compétence

II.1.2.2. Spécification de la Génération de plan de guide pour le filtrage des compétences

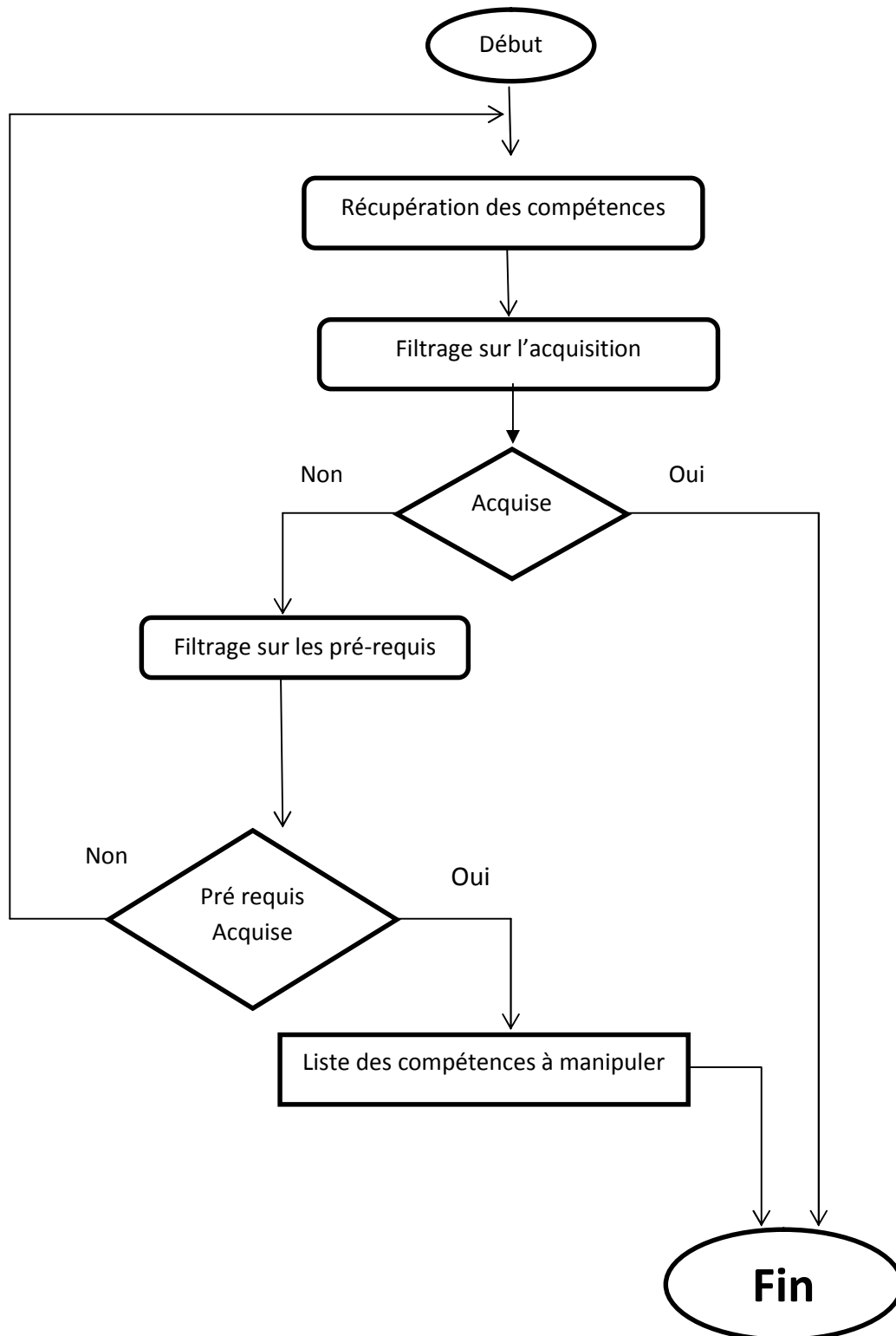


Figure II.3 : Spécification de la Génération de plan de guide pour le filtrage des compétences

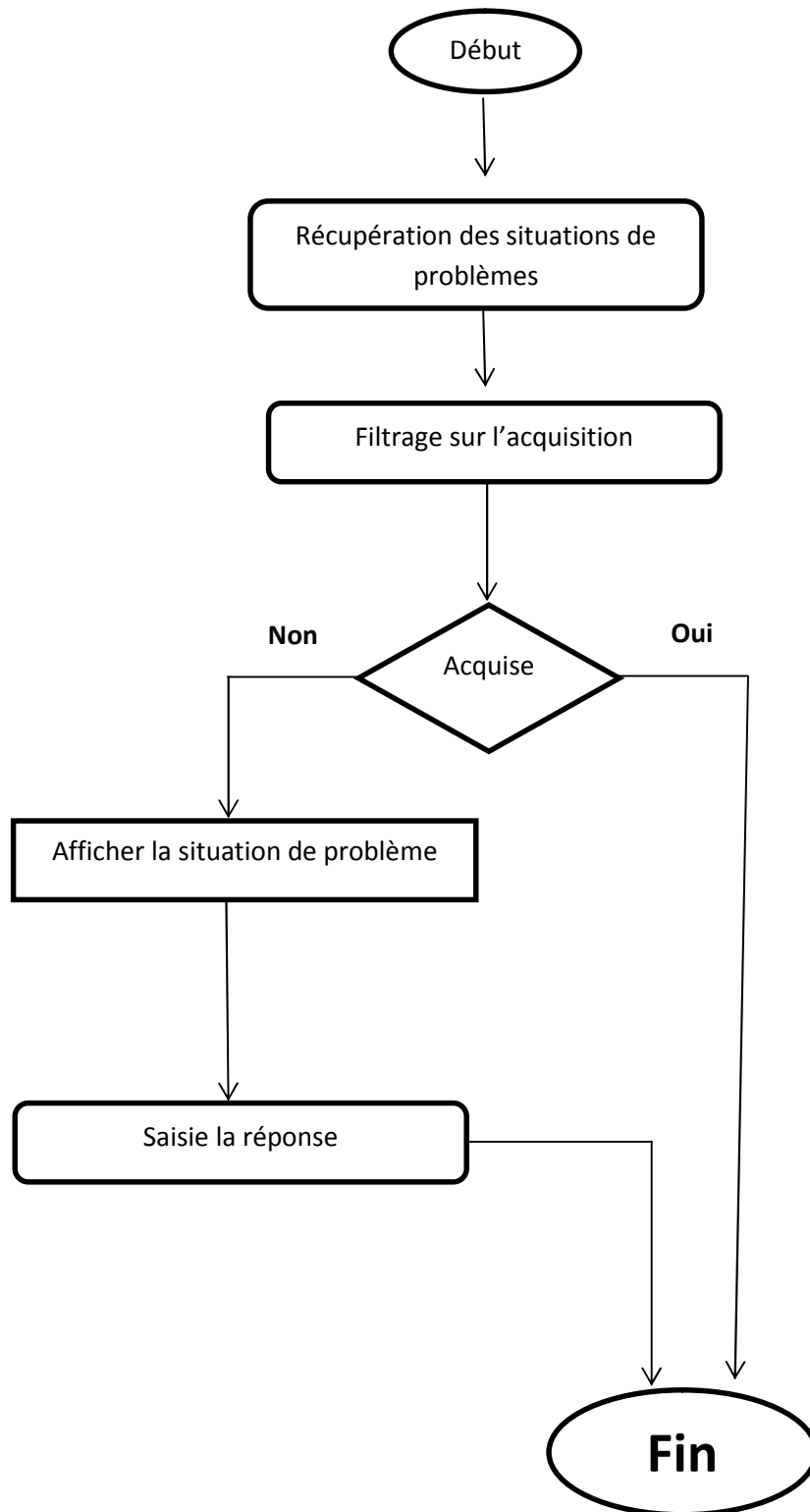
II.1.2.3. Spécification de la Génération de plan de guide pour le filtrage des situations de problèmes

Figure II.4 : Spécification de la Génération de plan de guide pour le filtrage des situations de problèmes

II.1.3 Analyse avec UML

II.1.3.1. Modèle des cas d'utilisation :

Détaillons les différentes étapes de la démarche que nous allons adopter afin d'aboutir au modèle des cas d'utilisation.

- Spécification des tâches et scénarios.
- Identification des cas d'utilisation.
- Ajout des relations entre cas d'utilisation.
- Finalisation des diagrammes de cas d'utilisation.

II.1.3.1.1. Spécification des tâches :

Les acteurs définis précédemment effectuent un certain nombre de tâches, ces tâches sont résumées dans le tableau ci-dessous :

Acteurs	Taches
Apprenant	T1 : Se connecter au site ; T2 : S'authentifier ; T3 : Choisir une compétence ; T4 : Résoudre une situation de problème ; T5 : Envoyer un message ; T6 : Se déconnecter.

Expert du domaine	<p>T7 : Se connecter au site ;</p> <p>T8 : S’authentifier ;</p> <p>T9 : Ajouter les compétences ;</p> <p>T10 : Modifier les compétences ;</p> <p>T11 : Supprimer les compétences ;</p> <p>T12: Ajouter des situations de problèmes et consignes.</p>
--------------------------	--

Figure II.5 : spécification des taches

II.1.3.1.2. Spécification des scénarios:

❖ **Définition des scénarios:**

Chaque résultat possible de l’exécution d’un cas d’utilisation est appelé scénario, c’est un chemin logique traversant le cas d’utilisation. Un scénario est une instance de cas d’utilisation. Un ensemble de scénarios pour un cas d’utilisation identifie tout ce qu’il peut arriver lorsque ce cas d’utilisation est en œuvre.

Acteur	Taches	Scénarios
Apprenant	T1 : se connecter au site	S1 : Saisir URL dans le navigateur.
	T2 :s’authentifier	<p>S2 : Choisir espace apprenant ;</p> <p>S3 : Saisir le pseudo et le mot de passe ;</p> <p>S4 : Cliquer sur le bouton «Connexion».</p>

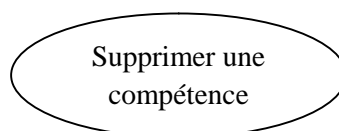
	T3 : choisir une compétence	S5 : Cliquer sur une compétence générée dans le plan guide adaptatif;
	T4 : résoudre une situation de problème	S6 : Saisir la réponse dans «champ de saisie» ; S7 : Cliquer sur «Enregistrer»
	T5 : envoyer un message	S8 : Saisir un message dans « Ecrivez nous » ; S9 : Cliquer sur « Envoyer »
	T6 : se déconnecter	S10 : Cliquer sur le bouton «Déconnexion»
Expert du domaine	T7: se connecter au site	S11 : Saisir URL dans le navigateur ;
	T8 : s'authentifier	S12 : Choisir espace expert du domaine S13 : Saisir le pseudo et le mot de passe ; S14 : Cliquer sur «Connexion».
	T9: ajouter les compétences	S15 : Cliquer sur «mise à jour compétence» ; S16 : Cliquer sur «ajouter» ; S17 : Remplir les champs ; S18 : Cliquer sur «enregistrer».
	T10 : modifier les compétences	S19 : Cliquer sur «mise à jour compétence» ; S20 : Cliquer sur «modifier» ; S21 : Modifier la compétence ; S22: Cliquer sur «enregistrer».

	T11: supprimer les compétences	S23 : Cliquer sur «mise à jour compétence» ; S24: Cliquer sur «supprimer».
	T12: Ajouter des situations de problèmes et consignes.	S25 : Cliquer sur «mise à jour situation problème » ; S26 : Cliquer sur « type de compétence » ; S27: Remplir les champs ; S28 : Cliquer sur « Enregistrer ».

Figure II.6 : spécification des scénarios

II.1.3.1.3. Identification et description des cas d'utilisation :

Les cas d'utilisation ou *use cases*. Un cas d'utilisation modélise un dialogue entre un acteur et le système. C'est la représentation d'une fonctionnalité offerte par le système. L'ensemble des cas d'utilisation forme toutes les façons possibles d'utilisation du système. Dans UML, le cas d'utilisation est représenté par un ovale comme suit:



On note que chaque cas d'utilisation doit avoir un objectif en soi, et pouvoir être réalisé indépendamment des autres.

Pour chaque acteur identifié précédemment, nous allons rechercher les différentes intentions « métier » selon lesquelles il utilise le système.

L'ensemble des cas d'utilisation est le suivant :

Pour l'apprenant :

- ✓ s'authentifier
- ✓ choisir une compétence
- ✓ résoudre une situation de problème
- ✓ envoyer un message
- ✓ se déconnecter

Pour l'expert de domaine :

- ✓ s'authentifier
- ✓ ajouter les compétences
- ✓ modifier les compétences
- ✓ supprimer les compétences
- ✓ ajouter les situations de problèmes et ses consignes
- ✓ se déconnecter

On obtient un diagramme préliminaire (voir **Figure II.3**), en représentant sur un schéma les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs. Un cas d'utilisation doit être relié au moins à un acteur.

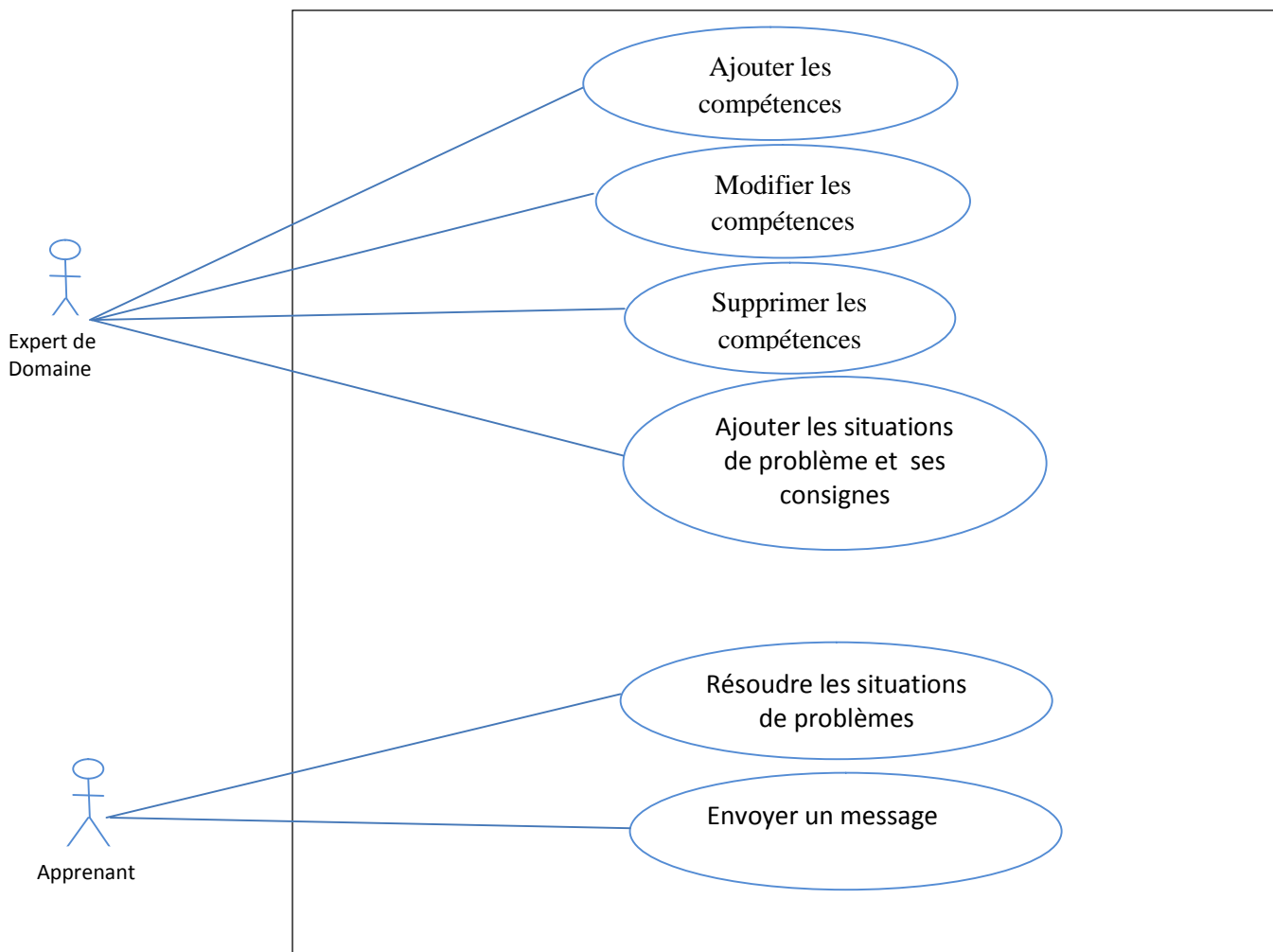


Figure II.7 : Diagramme préliminaire des cas d'utilisation

II.1.3.1.4. Ajout des relations entre cas d'utilisation :

Affinement du modèle de cas d'utilisation :

Pour affiner le diagramme de cas d'utilisation, nous allons utiliser trois types de relations standardisées entre cas d'utilisation qu'UML : une relation d'inclusion, une relation d'extension, une relation de généralisation/spécification.

On utilise la relation de généralisation/spécialisation afin de modéliser :

- Les différentes possibilités d'affichage de compétences sont : affichage de compétences générales, affichage de compétences composées et affichage de compétences élémentaires.
- Les deux types de comptes qu'on peut créer sont : compte apprenant et compte expert de domaine.

On utilise la relation d'extension afin de modéliser :

- L'accès de l'apprenant à son espace peut donner à faire afficher des compétences qui ont pas de pré requis, résoudre une situation de problème et envoyer un message.
- L'accès de l'expert de domaine à son espace peut donner lieu à l'ajout d'une compétence, modification d'une compétence, suppression d'une compétence et l'ajout des situations de problèmes et ses consignes.

Enfin, on pourrait relier les cas d'utilisations par des relations d'inclusion.

- Nous créons un cas d'utilisation que nous appelons « Authentification », qui doit être réalisé afin de permettre à ces acteurs d'exécuter leur propres cas d'utilisation.

II.1.3.1.5. Finalisation des diagrammes de cas d'utilisation :

Les cas d'utilisation constituent une technique qui permet de déterminer les besoins des utilisateurs et de capturer les exigences fonctionnelles d'un système. En d'autres termes ils décrivent le comportement d'un système de point de vue de ses utilisateurs. Ils décrivent les interactions entre les utilisateurs d'un système et le système lui-même.

Les cas d'utilisation peuvent être structurés. En plus de la relation de communication, qui consiste au déclenchement d'un cas d'utilisation par un acteur, les deux types de liens ou relations qui sont les plus utilisés : le lien d'utilisation et le lien d'extension.

- **Le lien d'utilisation** : Il indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.
- **Le lien d'extension** : Il indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions.

A) Diagramme des cas d'utilisation pour l'expert de domaine :

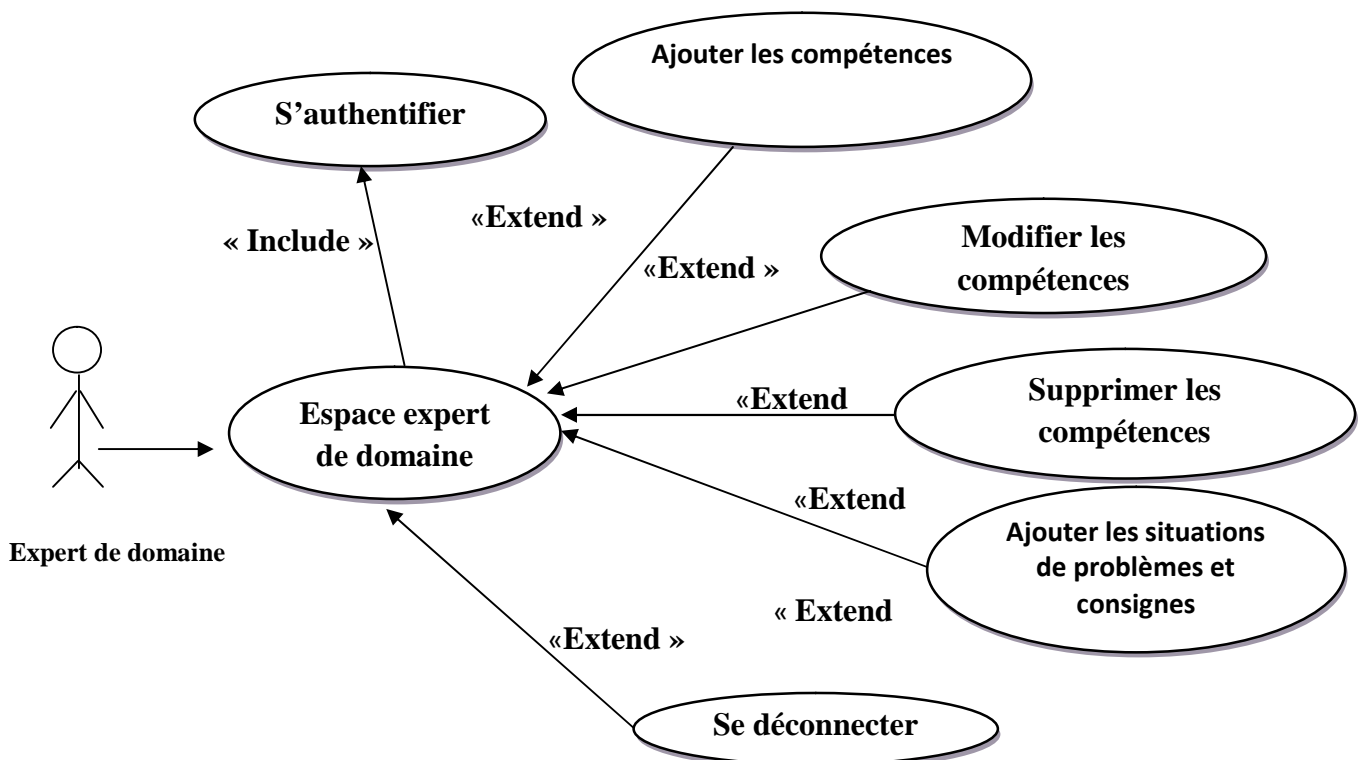


Figure II.8: Diagramme des cas d'utilisation «Expert de domaine»

B) Diagramme de cas d'utilisation pour l'apprenant :

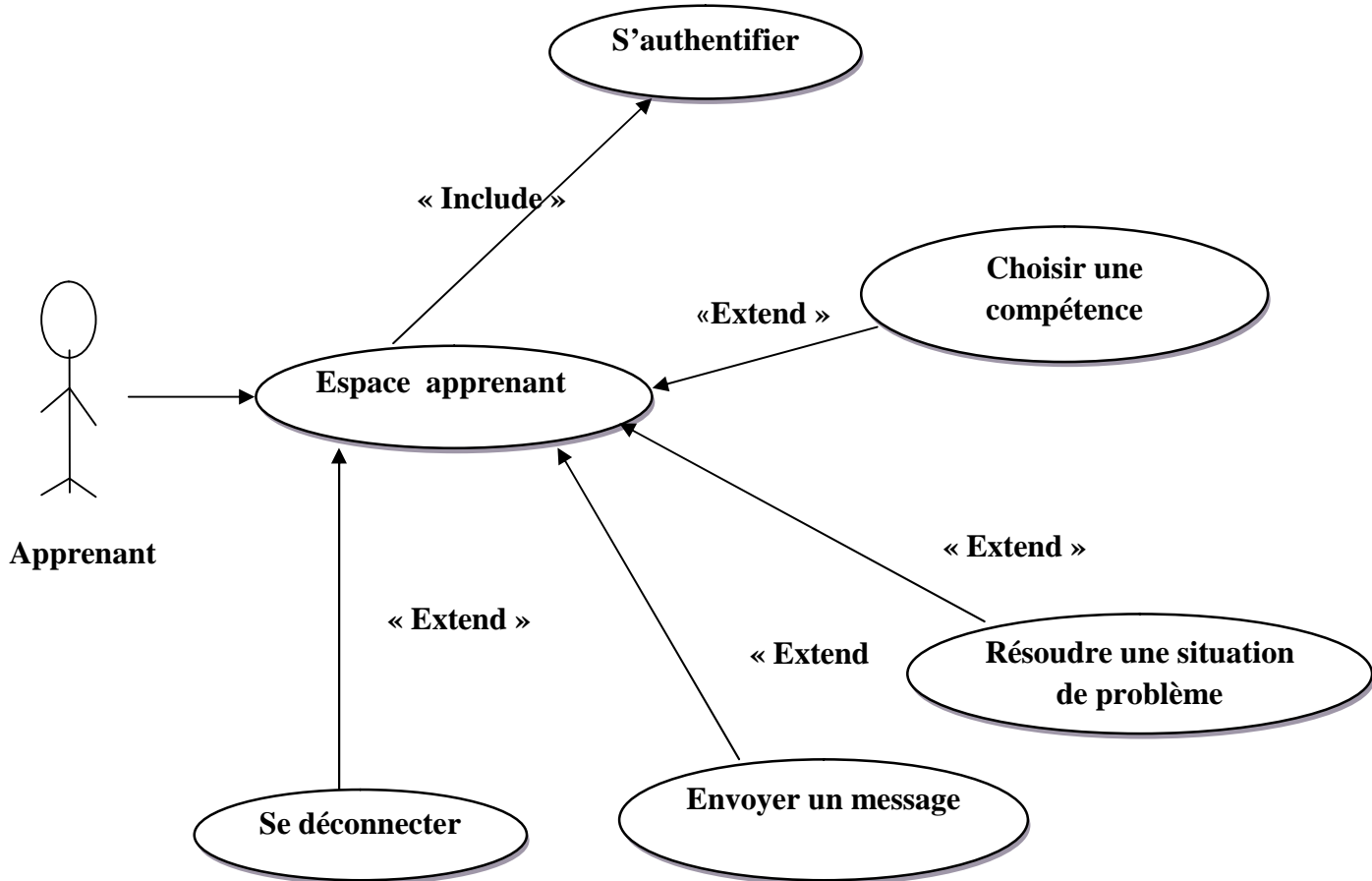


Figure II.9 : Diagramme de cas d'utilisation « Apprenant »

II.2. Conception avec UML

La conception est la phase la plus complexe du projet, surtout avec la fréquence des évolutions chronologiques. Dans cette phase une nouvelle vue du modèle fait son apparition. Cette vue exprime les modules et les exécutable physiques sans aller à la réalisation concrète du système.

La conception d'application web :

La conception d'application Web se distingue d'autres systèmes par deux activités majeurs, à savoir la répartition des objets sur le client et serveur et la définition de l'interface utilisateur sous forme de pages Web. Une application web implique l'existence d'au moins trois composants d'architecture significatifs :

- Le navigateur client ;
- Le serveur web ;
- Le serveur d'application.

II.2.1. Elaboration des diagrammes de séquence:

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée, lancer un traitement ; il met en évidence les objets manipulés ainsi que les opérations qui font passer d'un objet à l'autre. Dans notre cas on s'intéresse seulement à effectuer la représentation du diagramme de séquence pour quelques cas d'utilisation.

➤ Diagramme de séquence de réalisation de cas d'utilisation «s'authentifier»

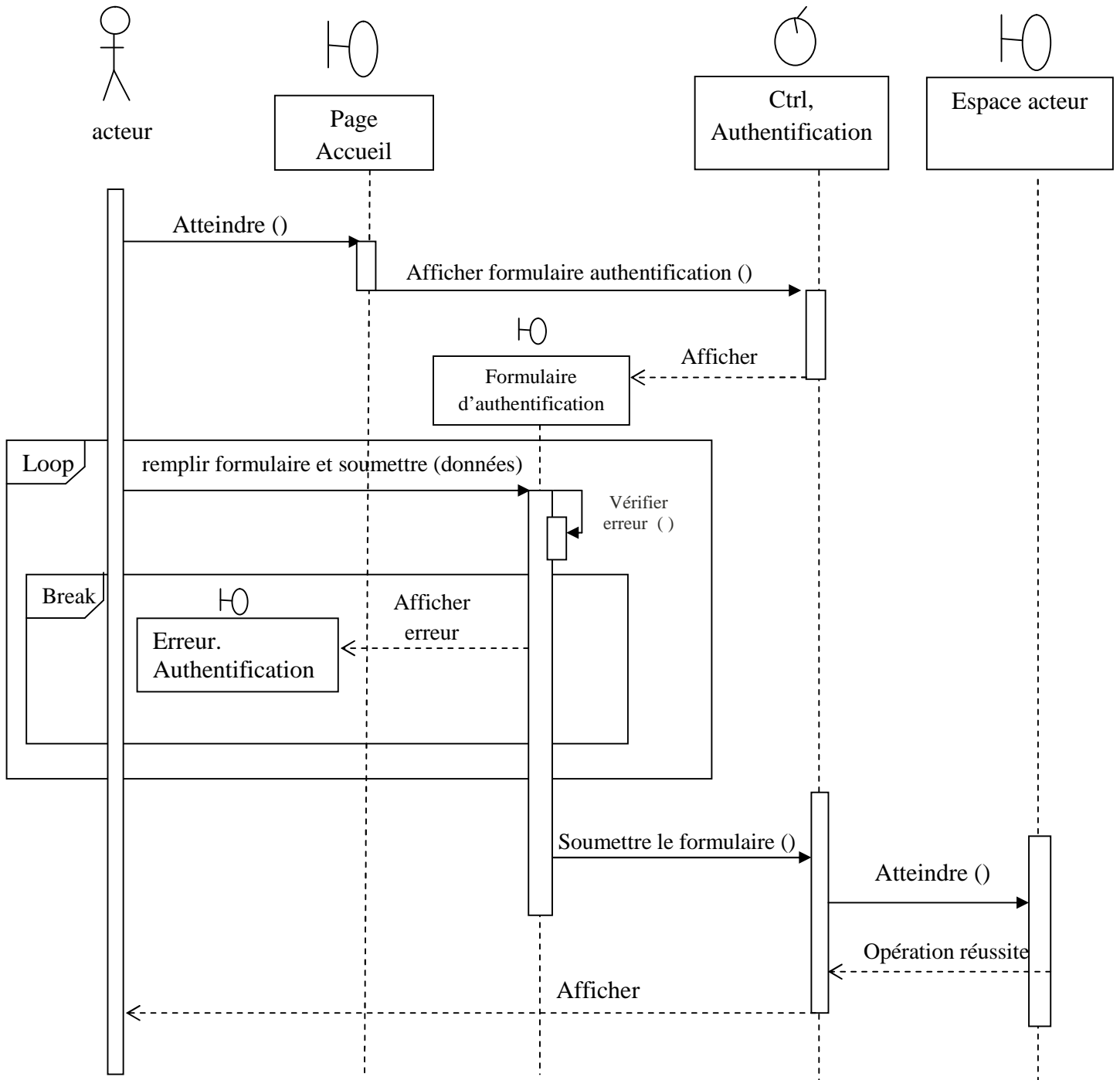


Figure II.10: Diagramme de séquence du cas d'utilisation « s'authentifier »

- Diagramme de séquence de réalisation de cas d'utilisation «résoudre une situation de problème »

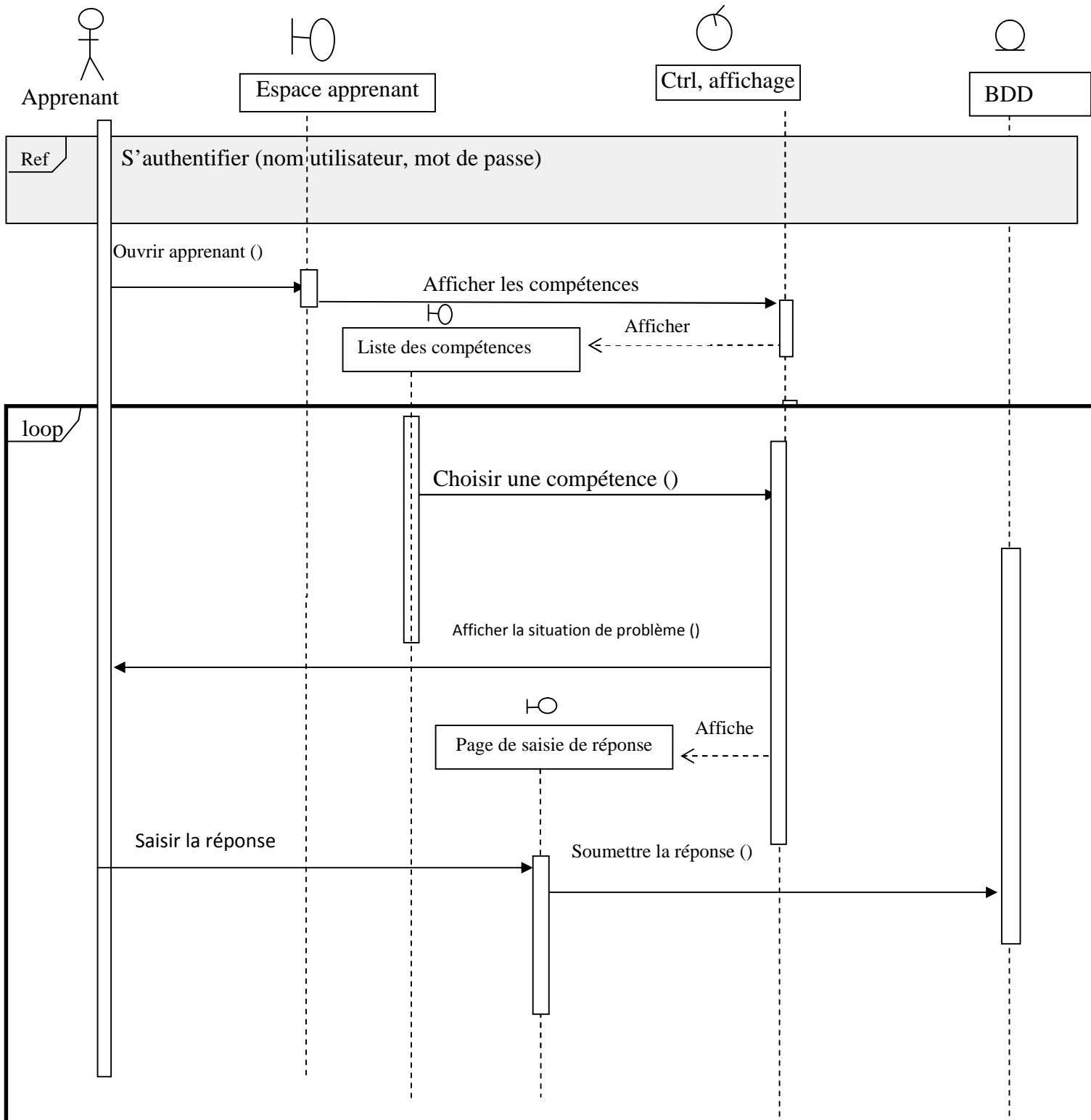


Figure II.11 : Diagramme de séquence du cas d'utilisation « résoudre une situation de problème ».

II.2.2. Diagrammes d'activité :

- Diagramme d'activité pour le cas d'utilisation : «s'authentifier».

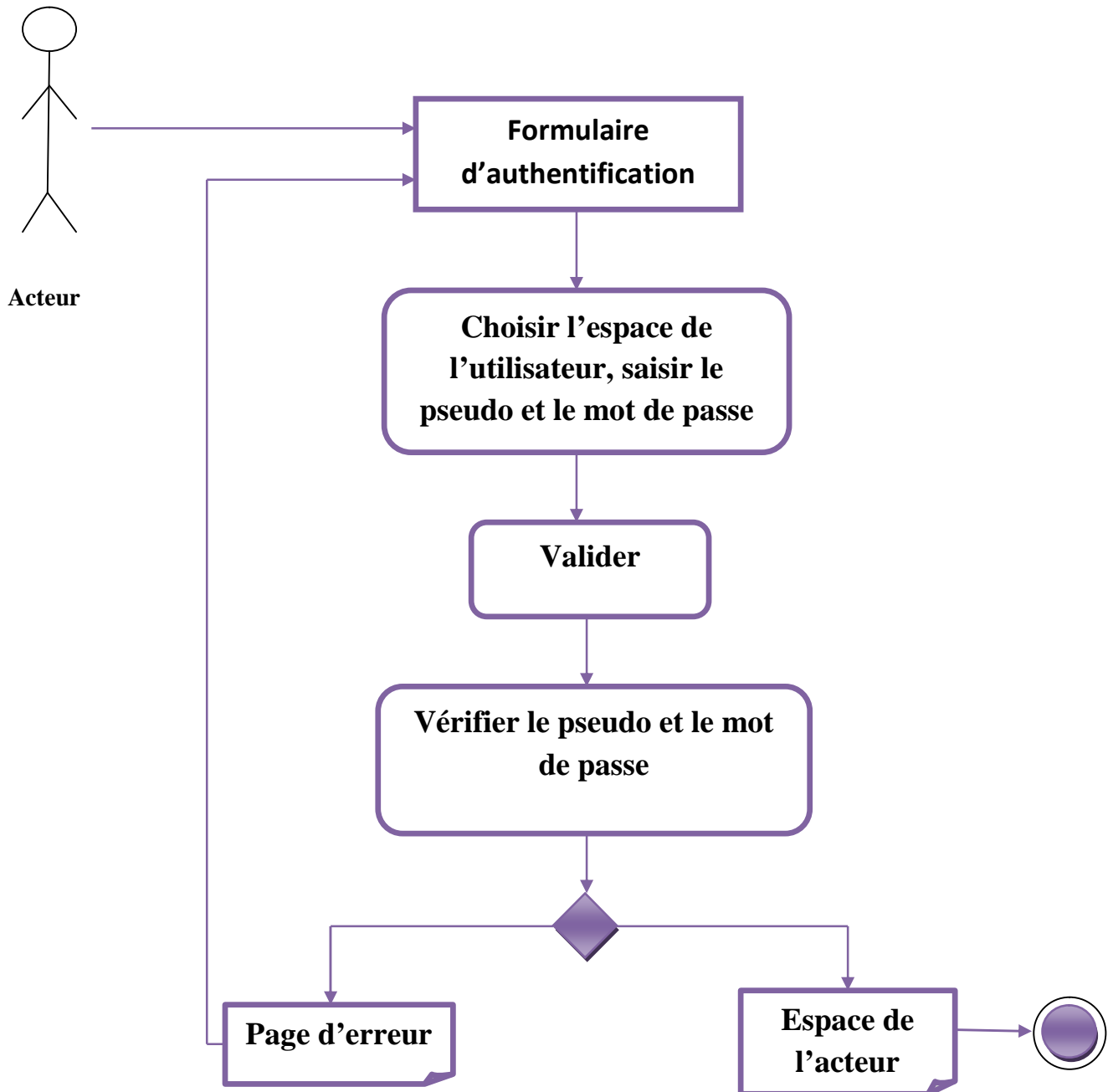


Figure II .12: Diagramme d'activité pour le cas d'utilisation «s'authentifier».

- Diagramme d'activité pour le cas d'utilisation «Répondre à une situation de problème»

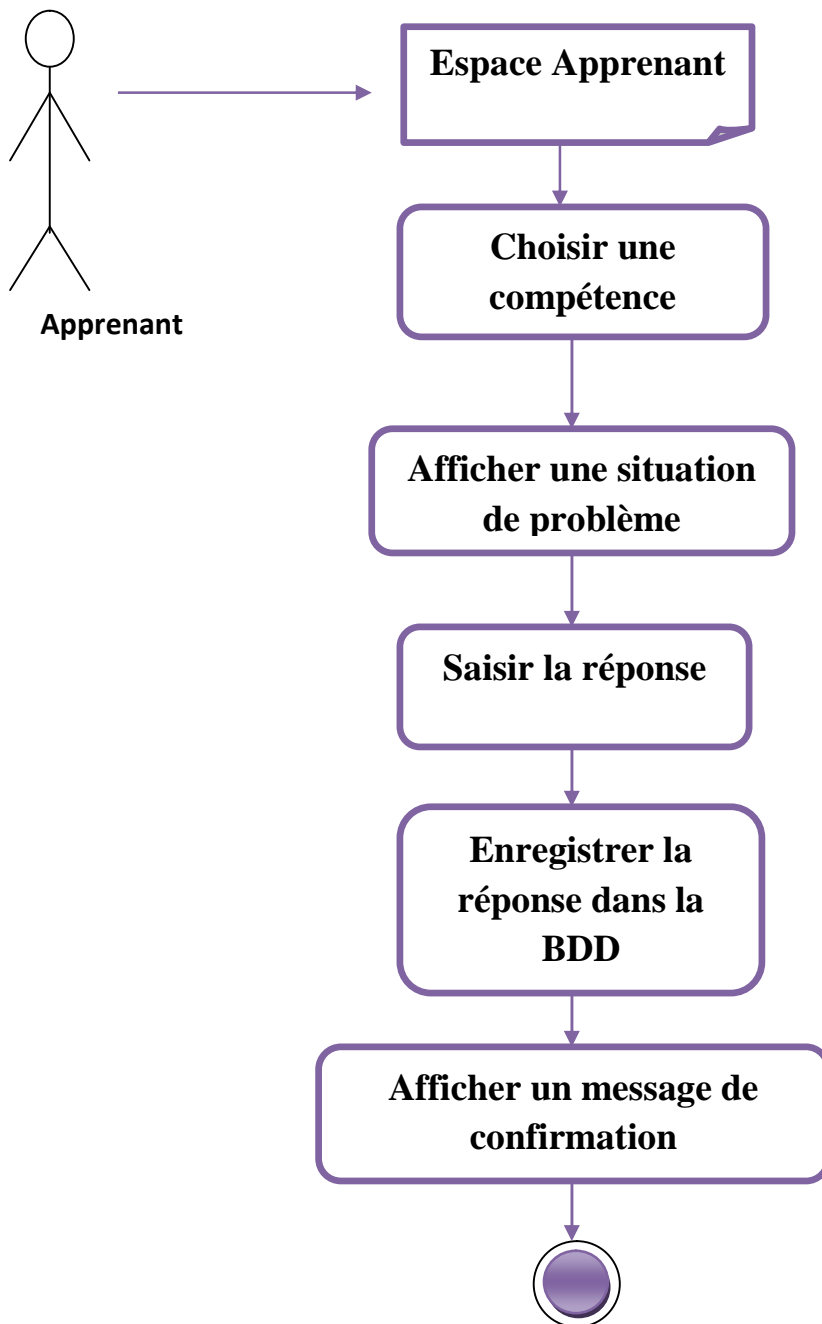


Figure II.13 : Diagramme d'activité pour le cas d'utilisation «Répondre à une situation de problème».

II .2.3. Diagramme de classe :

Les diagrammes de classes représentent la structure statique d'un système.

Il contient principalement des classes ainsi que leurs associations, mais on peut aussi y trouver des objets. L'intérêt majeur du diagramme de classe est de modéliser les entités du système.

Dans ce qui suit, nous allons présenter des diagrammes de classes de quelques cas d'utilisation

- Diagramme de classe général du cas d'utilisation «Authentification» :

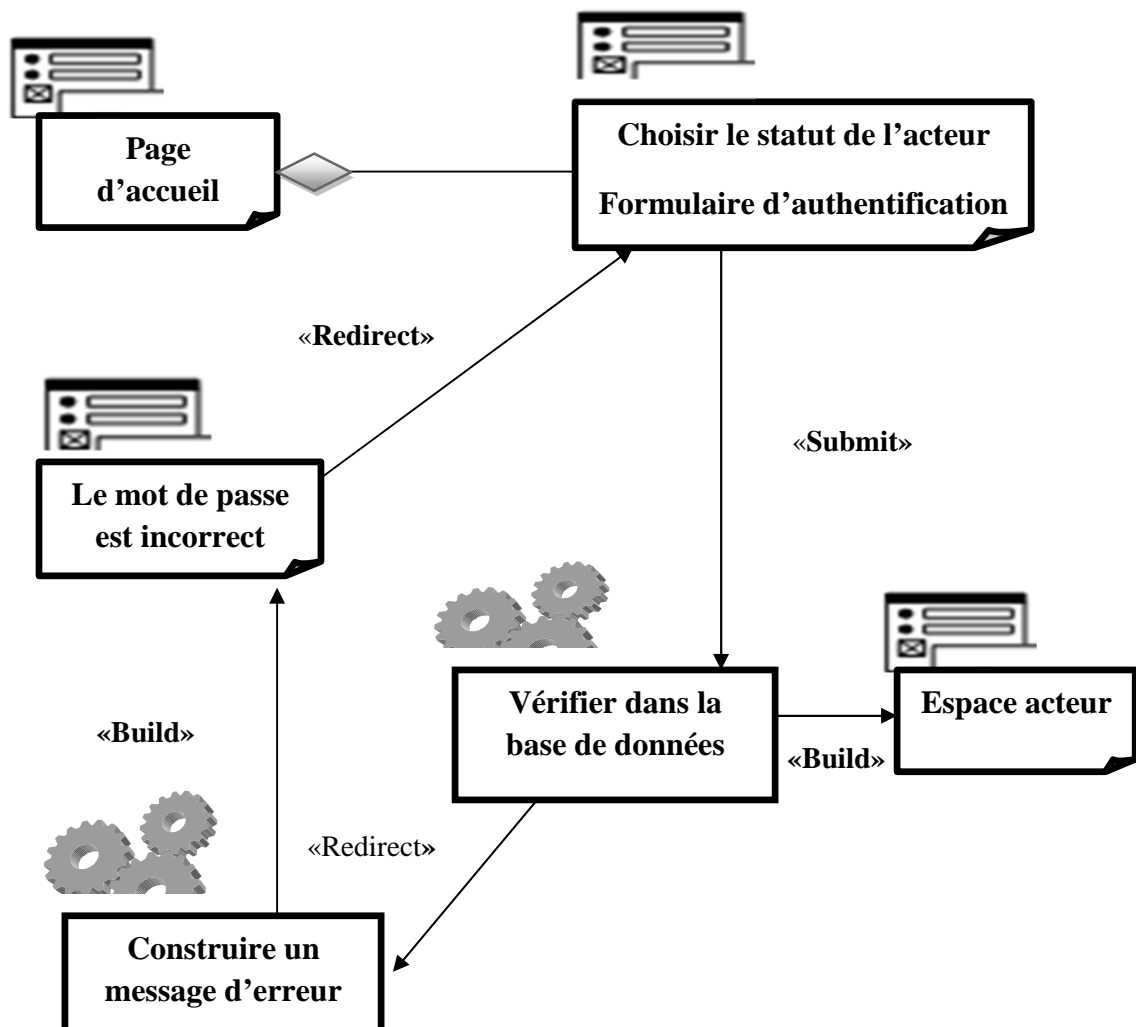


Figure II.14 : Diagramme de classe générale de cas d'utilisation «Authentification».

- Diagramme de classe général de cas d'utilisation «Résoudre une situation de problème».

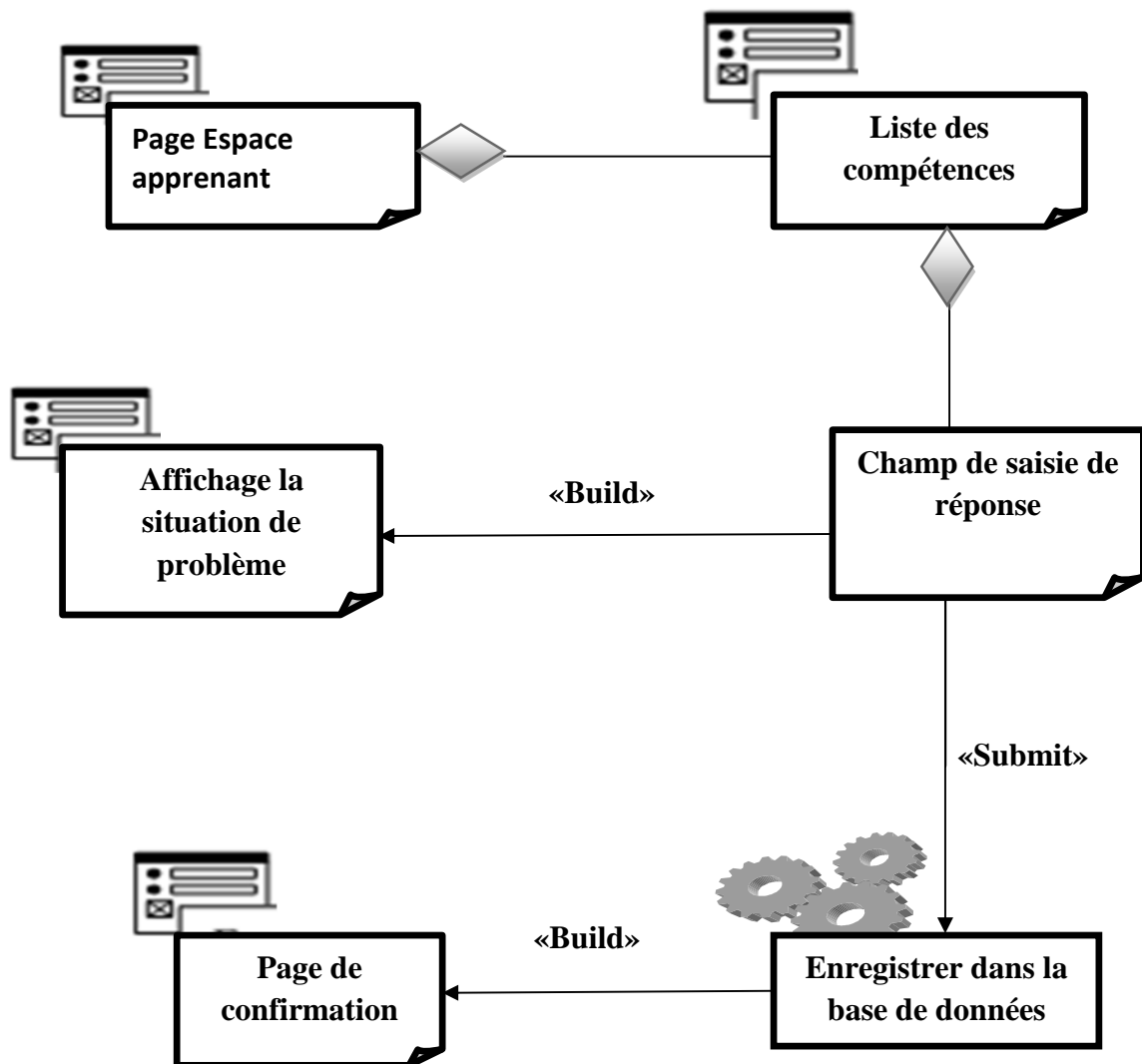


Figure II.15: Diagramme de classe général de cas d'utilisation «Résoudre une situation de problème».

➤ Diagramme de classes global

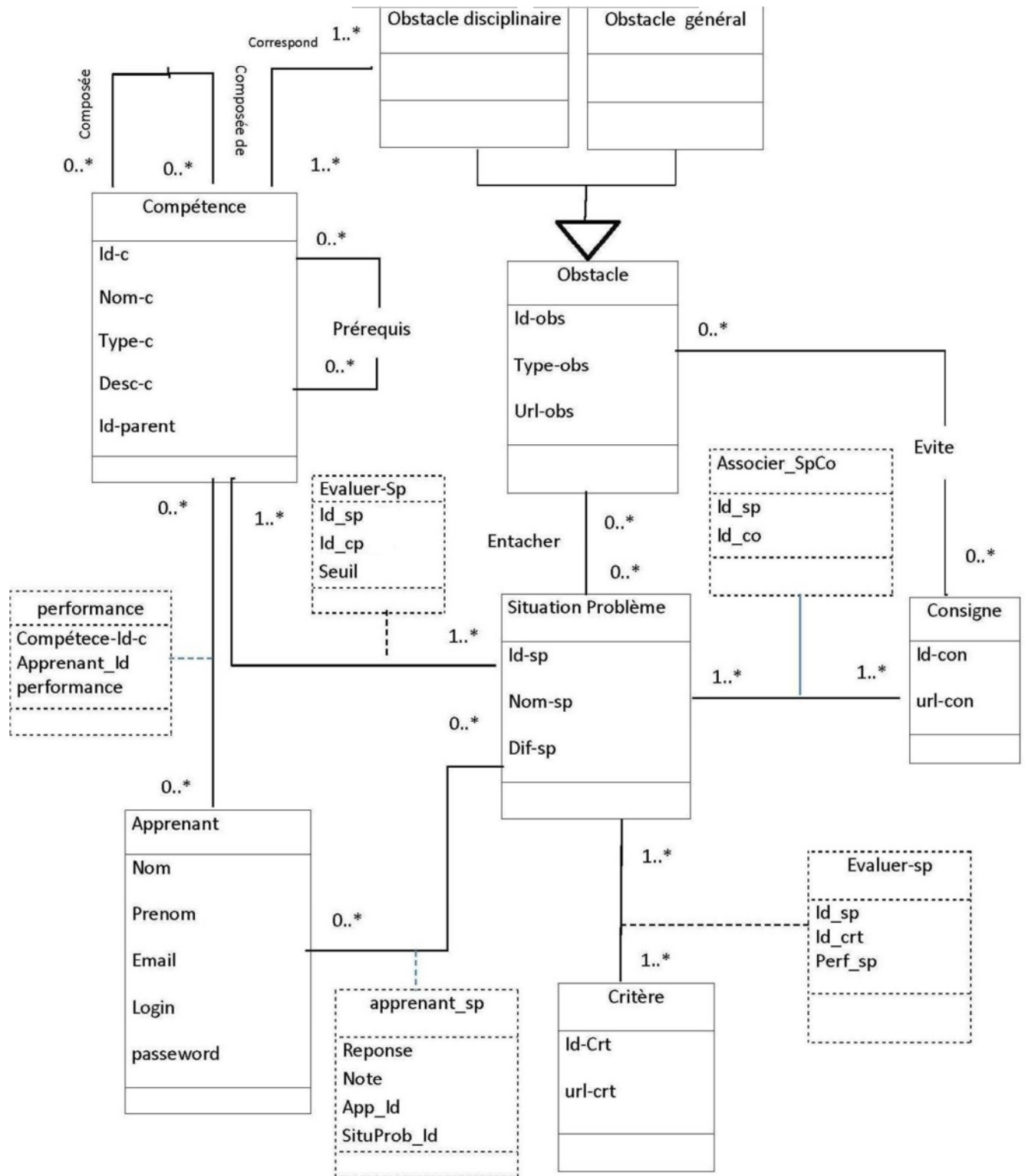


Figure II.15: Diagramme de classe global

II .2.4. Le modèle physique des données

Ce modèle nous donne la représentation physique de l'ensemble des tables de la base des données du système étudié.

❖ **définition :**

- **clé primaire** : identifie d'une façon unique une ligne dans une table ;
- **clé étrangère** : dépend d'une clé primaire située dans une autre table et migrée à partir de cette table.

❖ **Structure des tables :**

➤ **Table apprenant**

Nom de champ	Description	Type	clé
Id	Identifiant de l'apprenant	int(10)	Primaire
Nom	Nom de l'apprenant	varchar(45)	
Prénom	Prénom de l'apprenant	varchar(45)	
Email	Email de l'apprenant	varchar(80)	
Login	Login apprenant	varchar(20)	
Password	Mot de passe apprenant	varchar(20)	

➤ **Table Competence**

Nom de champ	Description	Type	Clé
Id_c	Identifiant compétence	int(10)	Primaire
Nom_c	Nom de compétence	varchar(100)	
Type	Type de compétence	varchar(100)	
Desc_c	Description de compétence	varchar(100)	
Id_c_parent	Identifiant de la compétence mère	int(10)	étrangère

➤ **Table prérequis**

Nom de champ	Description	Type	Clé
Competence_Id	Identifiant compétence	int(10)	étrangère
Competence_Id_pr	Identifiant compétence	int(10)	étrangère

➤ **Table situation problème**

Nom de champ	Description	Type	Clé
Id_sp	Identifiant situation de problème	int(10)	Primaire
Nom_sp	Nom situation problème	varchar(150)	
Dif_sp	Enoncée de situation de problème	text	

➤ **Table apprenant_sp**

Nom de champ	Description	Type	clé
Réponse	Réponse de la situation de problème	text	
Note	Note de la réponse	int(11)	
App_Id	Identifiant apprenant	int(10)	étrangère
SituProb_Id	Identifiant situation de problème	int(10)	étrangère

➤ **Table performance**

Nom de champ	Description	Type	clé
Compétence_Id_c	Identifiant de compétence	int(10)	étrangère
Apprenant_Id	Identifiant de l'apprenant	int(10)	étrangère
Performance	Résultat obtenu	Int(10)	

➤ **Table associer_spc**

Nom de champ	Description	Type	Clé
Id_sp	Identifiant situation de problème	int(10)	étrangère
Id_co	Identifiant consigne	int(10)	étrangère

➤ **Table consigne**

Nom de champ	Description	Type	Clé
Id_co	Identifiant consigne	int(10)	Primaire
Contenu	Contenu de consigne	text	

➤ **Table evaluer_cpsp**

Nom de champ	Description	Type	Clé
Id_sp	Identifiant situation de problème	int(10)	étrangère
Id_cp	Identifiant compétence	int(10)	étrangère
Seuil	Seuil de la situation de problème	varchar(10)	

➤ **Table critere**

Nom de champ	Description	Type	Clé
Id_crt	Identifiant critère	int(10)	Primaire
Url_crt	Contenu de la critère	varchar(100)	

➤ **Table entacher**

Nom de champ	Description	Type	Clé
Id_sp	Identifiant situation de problème	int(10)	étrangère
Id_ob	Identifiant obstacle	int(10)	étrangère

➤ **Table evaluer_sp**

Nom de champ	Description	Type	Clé
Id_sp	Identifiant situation de problème	int(10)	étrangère
Id_crt	Identifiant critère	int(10)	étrangère
Perf_sp	Performance de situation de problème	varchar(100)	

➤ **Table eviter_coob**

Nom de champ	Description	Type	Clé
Id_ob	Identifiant obstacle	int(10)	étrangère
Id_co	Identifiant consigne	int(10)	étrangère

➤ **Table obstacle**

Nom de champ	Description	Type	Clé
Id_ob	Identifiant obstacle	int(10)	Primaire
Type_ob	Type d'obstacle	varchar(100)	
Url_ob	L'url d'obstacle	varchar(150)	

Conclusion

Dans ce chapitre nous avons proposé une démarche de modélisation pour développer notre application, cette démarche est basée sur l'UML. Nous avons commencé par l'identification des acteurs, et spécifications des tâches, puis l'architecture globale du système en spécifiant ses fonctions. En phase de conception, nous avons élaboré les diagrammes de séquence, les diagrammes d'activité et les diagrammes des classes, et pour conclure ce chapitre nous avons défini les tables de notre base de données.

Le chapitre suivant sera consacré à la partie réalisation de notre application et en définissant les outils et l'environnement de développement utilisés.

Chapitre III :
Réalisation

Introduction

Après avoir présenté dans le chapitre précédent la conception et le fonctionnement global de notre système, nous arrivons dans ce chapitre à la mise en œuvre de notre application.

Ce chapitre sera divisé en deux parties : la première partie sera consacrée à la description des outils de développement de notre application. La deuxième partie contiendra les détails techniques d'implémentation des différentes fonctionnalités offertes par notre système.

III.1. Outils utilisés :

III.1.1. Langages à balises :

➤ HTML :

HTML est un langage de description (et non pas un langage de programmation) qui permet de décrire l'aspect d'un document, d'y inclure des informations variées (texte, images, son, animations, etc.) et d'établir des relations cohérentes entre ces informations grâce aux liens hypertextes.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="Candidat_MasterPage" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
</head>
<body>
<form id="form1" runat="server">
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
<div id="Entete">
<h2 style="padding: 0px 5px 5px 5px; margin: 0px 5px 5px 5px; font-family: Arial, Helvetica, sans-serif; color: #FFFFFF; width: 50%; float:left">
<div id="connexion" align="right">
<%--<asp:HyperLink ID="L" runat="server" ForeColor="White"
NavigateUrl="~/Connexion.aspx"></asp:HyperLink--%>
<asp:LinkButton ID="lnkDeconnecter" runat="server" ForeColor="White"
Text="Déconnecter" onclick="lnkDeconnecter_Click">
</asp:LinkButton>
<br /> <br /><br />
<table>
<tr>
<td>
<asp:Image ID="Image2" runat="server" Width="20" Height="20"
BorderStyle="None" />
</td>
<td>
<asp:LinkButton ID="LkNom" runat="server" ForeColor="White"
Text="" CssClass="formatSerif" onclick="LkNom_Click">
</asp:LinkButton>
</td>
</tr>
</table>
```

Figure III.1 : Exemple de fichier.html.

➤ **les feuilles de style (CSS Cascading stylesheets):**

Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation. En effet le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent.

```
body
{
  padding: 0px;
  margin: 0px;
  background-color: #eeeeee;
  width: 100%;
}
div#Entete
{
  border-width: thin;
  border-color: #000000;
  width: 1000px;
  height: 100px;
  padding: 0px;
  margin: 0px auto 0px auto;
  background-image: url('../Image/serch-bg.jpg');
  border-right-style: groove;
  border-left-style: groove;
}
div#connexion
{
  width: 40%;
  padding: 2px;
  margin: 0px;
  float: right;
}
div#Main
{
  border: thin groove #000000;
  width: 1000px;
  height: 1200px;
  padding: 0px;
  margin: 0px auto 0px auto;
  background-color:white;
  min-height:100%;
}
}
```

Figure III.2 : Exemple de code style.css.

III.1.2. Langages de script:

➤ **jQuery :**

jQuery est une extension de JavaScript, il permet aux développeurs web d'ajouter des fonctionnalités supplémentaires à leurs sites Web. Il est open source et disponible gratuitement sous la licence MIT. Au cours des dernières années, jQuery est devenue la bibliothèque JavaScript la plus populaire utilisée dans le développement web .

Pour mettre en œuvre jQuery, un développeur web doit simplement faire référence au fichier JavaScript jQuery dans le code HTML d'une page Web . Certains sites Web hébergent leur propre copie locale de jQuery, tandis que d'autres simplement référence à la Bibliothèque hébergé par Google ou le serveur jQuery. Par exemple, une page Web peut charger la bibliothèque jQuery en utilisant la ligne suivante dans la section <head> du HTML:

```
<Script type= " text \ javascript "  
src = " / / ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"> </ script>
```

Une fois la bibliothèque jQuery est chargée, une page Web peut appeler n'importe quel jQuery fonction soutenue par la bibliothèque. Les exemples communs incluent la modification du texte, le traitement des données de formulaire, des éléments sur une page en mouvement, et l'exécution des animations. jQuery peut également fonctionner avec Ajax code et langages de script tels que ASP et PHP pour accéder aux données d'une base de données . Depuis jQuery fonctionne sur le côté client (plutôt que le serveur web), il peut mettre à jour les informations sur une page Web en temps réel , sans avoir à recharger la page. Un exemple courant est "autocomplete", dans lequel au fur et a mesure ou on saisie des données dans un formulaire, grâce à jQuery, il est possible de suggérer la suite des mots tapés présent dans une base de donné.

Outre sa licence libre, l'autre raison principale jQuery a acquis une telle popularité est son cross-browser compatible. Étant donné que chaque navigateur affiche HTML, CSS et JavaScript différemment, il peut être difficile pour un développeur web pour faire apparaître un site web la même dans tous les navigateurs . Au lieu d'avoir à écrire des fonctions personnalisées pour chaque navigateur, un développeur Web peut utiliser une fonction de jQuery unique qui fonctionnera dans Chrome, Safari, Firefox et Internet Explorer. Ce support multi-navigateur a conduit de nombreux développeurs à passer du niveau javascript jQuery, car il simplifie considérablement le processus de codage.

```
<link href="jquery-ui.css" rel="stylesheet" type="text/css"/>
<script type="text/javascript" src="jquery.min.js"></script>
<script type="text/javascript" src="jquery-ui.min.js"></script>
<script type="text/javascript">
$(document).ready(function () {
    SearchText();
});
function SearchText() {
    $(".autosuggest").autocomplete({
        source: function (request, response) {
            $.ajax({
                type: "POST",
                contentType: "application/json; charset=utf-8",
                url: "Recherche.aspx/GetAutoCompleteData",
                data: '{"username":"' + document.getElementById('txtSearch').value + '"}',
                dataType: "json",
                success: function (data) {
                    response(data.d);
                },
                error: function (result) {
                    alert("Error");
                }
            });
        }
    });
};

```

Figure III.3 : Exemple de script jQuery.

III.1.3. Serveurs :

➤ Serveur Web :

Pour exécuter des applications Web, il faut disposer d'un serveur. Un serveur Web est un logiciel permettant à des clients d'accéder à des pages web, à partir d'un navigateur (aussi appelé browser). Un serveur Web est donc un simple logiciel capable d'interpréter les requêtes http arrivants sur le port associé au protocole http et de fournir une réponse avec ce même protocole.

Exemple de serveur web :

- Microsoft IIS (Internet Information Serveur)
- Apache
- Microsoft PWS (Personal Web Server)
- Notre application est gérée par le serveur Microsoft Apache qui est le serveur Web de Microsoft, il contrôle les pages Web (celles évidemment qui sont hébergées sur sa

machine) et qui envoie le HTML de ces pages Web aux navigateurs des clients qui en font la demande.

➤ **Serveur de base de données :**

Présentation de Microsoft SQL Server :

Microsoft SQL Server est un système de gestion de base de données (abrégé en SGBD ou SGBDR pour « Système de gestion de base de données relationnelles ») développé et commercialisé par la société Microsoft.

Bien qu'il ait été initialement développé par Sybase et Microsoft, Ashton-Tate a également été associé à sa première version, sortie en 1989. Cette version est sortie sur les plates-formes Unix et OS/2. Depuis, Microsoft a porté ce système de base de données sous Windows et il est désormais uniquement pris en charge par ce système.

En 1994, le partenariat entre les deux sociétés ayant été rompu, Microsoft a sorti la version 6.0 puis 6.5 seul, sur la plate-forme Windows NT.

- Microsoft a continué de commercialiser le moteur de base de données sous le nom de SQL Server
- Tandis que Sybase, pour éviter toute confusion, a renommé Sybase SQL Server en Sybase Adaptive Server Enterprise.

Microsoft SQL Server fait désormais partie de la stratégie technique de Microsoft en matière de base de données. Le moteur MSDE, qui est la base de SQL Server, doit à terme remplacer le moteur Jet (celui qui gère les bases Access) dans les applications telles qu'Exchange et Active Directory.

La version 2005 de SQL Server est sortie le 3 novembre 2005 en même temps que Visual Studio 2005. La prise en charge de Windows Vista de Windows Server 2008 n'a été ajoutée qu'à partir du Service Pack 2 (SP2).

La version 2008 de SQL Server (nom de code Katmai) est disponible depuis août 2008. Elle est actuellement au niveau de service pack 3. Elle est disponible en 9 langues, dont le français.

III.2. Environnement de développement

III.2.1. langage de programmation java:

Java est un langage de programmation récent (les premières versions datent de 1995) développé par Sun Microsystems. Il est fortement inspiré des langages C et C++. Comme C++, Java fait partie de la « grande famille » des langages orientés objets. Il répond donc aux trois principes fondamentaux de l'approche orientée objet (POO) : l'encapsulation, le polymorphisme et l'héritage.

Java a rapidement intéressé les développeurs pour quatre raisons principales :

- ✓ C'est un langage orienté objet dérivé du C, mais plus simple à utiliser et plus « pur » que le C++. On entend par « pur » le fait qu'en Java, on ne peut faire que de la programmation orienté objet contrairement au C++ qui reste un langage hybride, c'est-à-dire autorisant plusieurs styles de programmation. C++ est hybride pour assurer une compatibilité avec le C ;
- ✓ Il est doté, en standard, de bibliothèques de classes très riches comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, contrôles, menus, graphisme), la programmation multi-threads (multitâches), la gestion des exceptions, les accès aux fichiers et au réseau ... L'utilisation de ces bibliothèques facilite grandement la tâche du programmeur lors de la construction d'applications complexes ;
- ✓ Il est doté, en standard, d'un mécanisme de gestion des erreurs (les exceptions) très utile et très performant. Ce mécanisme, inexistant en C, existe en C++ sous la forme d'une extension au langage beaucoup moins simple à utiliser qu'en Java ;
- ✓ Il est multi plates-formes : les programmes tournent sans modification sur tous les environnements où Java existe (Windows, Unix et Mac) ;

Ce dernier point a contribué à utiliser d'abord Java pour développer des applets qui sont des applications pouvant être téléchargées via l'Internet et exécutées dans un navigateur sur n'importe quelle plate-forme. Ainsi, une page statique HTML peut s'enrichir de programmes qui lui donneront un comportement dynamique. Il possède deux particularités :

- **Multi plates-formes**

Java est un langage qui doit être compilé et interprété. Dans une première phase on compile un programme (un ou plusieurs fichiers source .java) en fichiers .class. Le compilateur génère un fichier .class pour chacune des classes définies dans le(s) fichier(s) .java. L'ensemble des fichiers .class est ensuite interprété par la Machine Virtuelle Java (Java Virtual Machine) pour exécuter le programme (seule la JVM dépend du système). Ce principe donne à Java une portabilité maximale (Windows, Unix, MacIntosh ...).

- **Applets et applications**

Il est possible de développer des applications isolées (standalone applications), fonctionnant avec l'interpréteur comme un programme habituel dans un langage classique mais aussi des "applets". Les applets sont des programmes qui peuvent être téléchargés sur l'Internet puis exécutés automatiquement quand ils sont intégrés à dans des pages HTML. Dans ce dernier cas, l'ensemble des fichiers .class est utilisé avec un fichier HTML qui fait appel à une des classes.

III.2.2. Architecture MVC:

L'organisation d'une interface graphique est délicate. L'architecture MVC ne prétend pas en éliminer tous les problèmes, mais fournit une première approche pour ce faire. Offrant un cadre normalisé pour structurer une application, elle facilite aussi le dialogue entre les concepteurs.

L'idée est de bien séparer les données, la présentation et les traitements. Il en résulte les trois parties énumérées plus haut : le modèle, la vue et le contrôleur.

➤ **Modèle**

Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de

données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation. Le modèle ne contient aucun lien direct vers le contrôleur ou la vue. Sa communication avec la vue s'effectue au travers du patron Observateur.

Le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les emplois du temps, le modèle peut avoir des méthodes pour avoir tous les cours d'une salle, tous les cours d'une personne ou tous les cours d'un groupe de TD.

➤ **Vue**

C'est avec quoi l'utilisateur interagit, se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, coché d'une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

Plusieurs vues peuvent afficher des informations partielles ou non d'un même modèle. Par exemple si une application de conversion de base a un entier comme unique donnée, ce même entier peut être affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir à l'utilisateur la possibilité de changer de vue.

➤ **Contrôleur**

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jours. D'après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est lui « observable ». Certains événements de l'utilisateur ne

concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

Par exemple, dans le cas d'une base de données gérant les emplois du temps des professeurs d'une école, une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser tous ses cours. Ceci ne modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne.

Quand un même objet contrôleur reçoit les événements de tous les composants, il lui faut déterminer quelle est l'origine de chaque événement. Ce tri des événements peut s'avérer fastidieux et peut conduire à un code peu élégant (un énorme switch). C'est pourquoi le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements d'une partie des composants.

➤ Flux de traitement

En résumé, lorsqu'un client envoie une requête à l'application :

- la requête envoyée depuis la vue est analysée par le contrôleur (par exemple un clic de souris pour lancer un traitement de données) ;
- le contrôleur demande au modèle approprié d'effectuer les traitements et notifie à la vue que la requête est traitée ("via" par exemple un handler ou callback) ;
- la vue notifiée fait une requête au modèle pour se mettre à jour (par exemple affiche le résultat du traitement "via" le modèle).

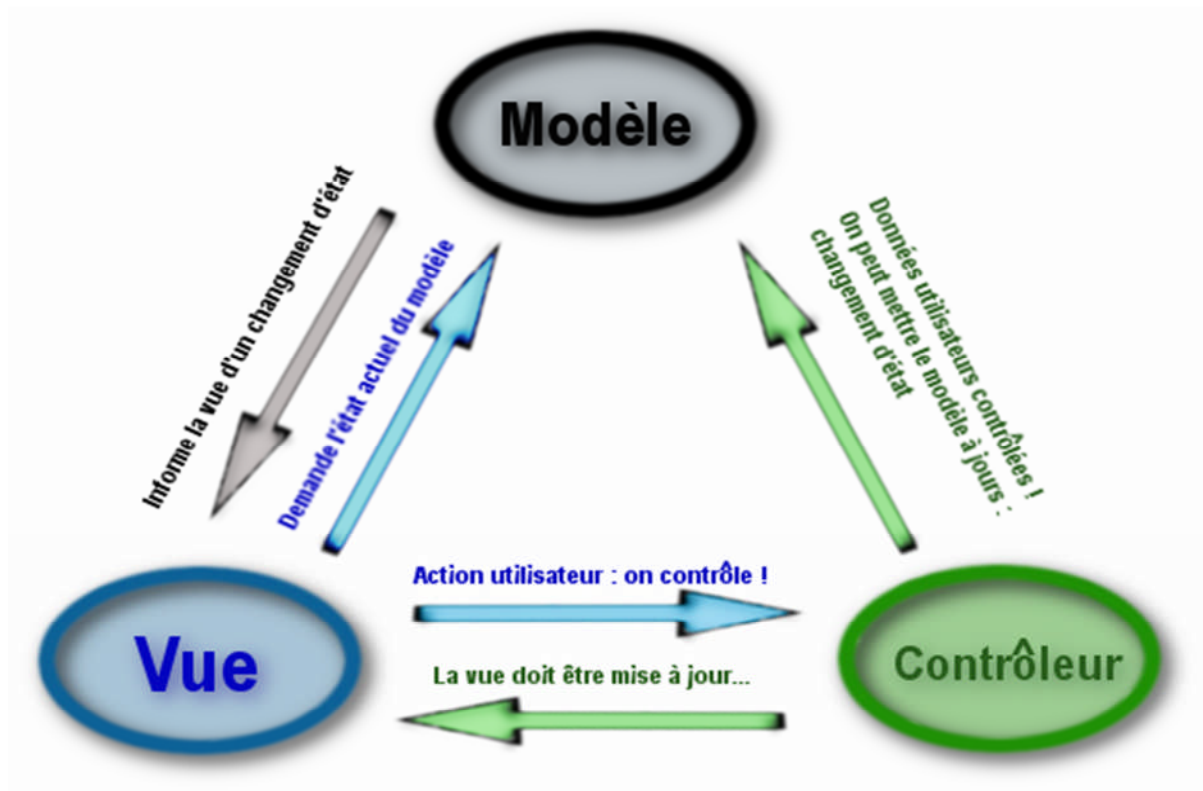


Figure III.4 : interface " pattern MVC "

III.2.3. Avantages de l'architecture MVC

Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées.

Le MVC montre ses limites dans le cadre des applications utilisant les technologies du web, bâties à partir de serveurs d'applications. Des couches supplémentaires sont alors introduites ainsi que les mécanismes d'inversion de contrôle et d'injection de dépendance.

III.2.4. Différence avec l'architecture trois tiers

L'architecture trois tiers est un modèle en couches, c'est-à-dire, que chaque couche communique seulement avec ses couches adjacentes (supérieures et inférieures) et le flux de contrôle traverse le système de haut en bas; les couches supérieures contrôlent les couches inférieures, c'est-à-dire, que les couches supérieures sont toujours sources d'interaction (clients) alors que les couches inférieures ne font que répondre à des requêtes (serveurs).

Dans le modèle MVC, il est généralement admis que la vue puisse consulter directement le modèle (lecture) sans passer par le contrôleur. Par contre, elle doit nécessairement passer par le contrôleur pour effectuer une modification (écriture). Ici, le flux de contrôle est inversé par rapport au modèle en couches, le contrôleur peut alors envoyer des requêtes à toutes les vues de manière à ce qu'elles se mettent à jour.

Dans l'architecture trois tiers, si une vue modifie les données, toutes les vues concernées par la modification doivent être mises à jour, d'où l'utilité de l'utilisation du MVC au niveau de la couche de présentation. La couche de présentation permet donc d'établir des règles du type « mettre à jour les vues concernant X si Y ou Z sont modifiés ». Mais ces règles deviennent rapidement trop nombreuses et ingérables si les relations logiques sont trop élevées. Dans ce cas, un simple rafraîchissement des vues à intervalle régulier permet de surmonter aisément ce problème. Il s'agit d'ailleurs de la solution la plus répandue en architecture trois tiers, l'utilisation du MVC étant moderne et encore marginale.

III.3. Les outils de développement:

III.3.1. IDE NetBeans:

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

L'IDE Netbeans s'enrichit à l'aide de plugins.

Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web, etc).

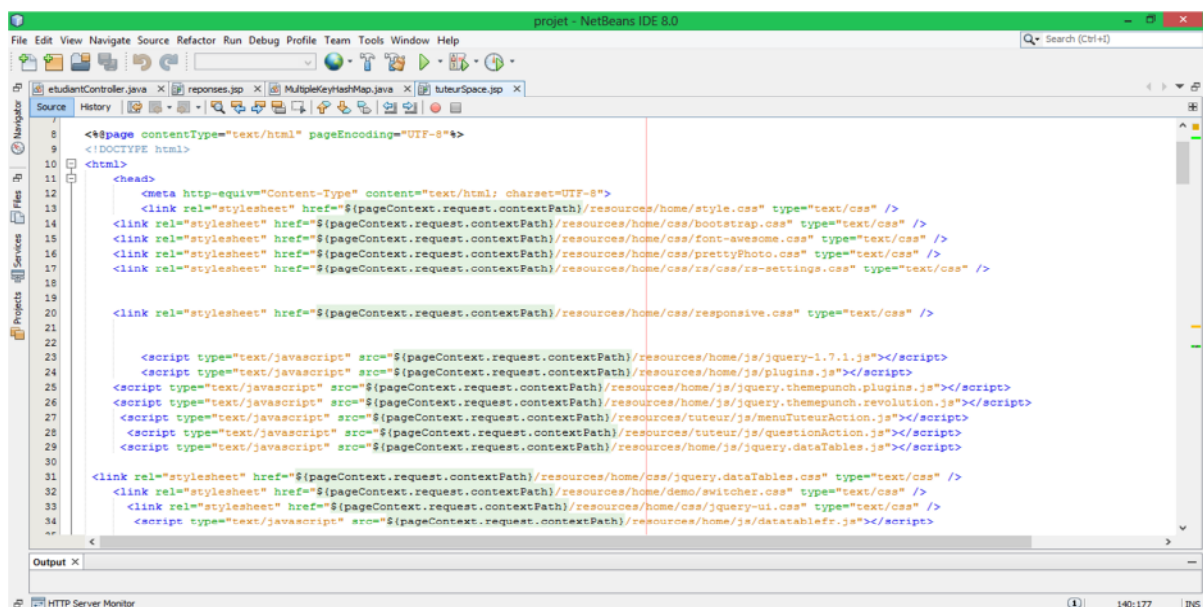


Figure III.5 : interface " NetBeans 8.0 "

III.3.2. Le SGBD (MYSQL 5) :

MySQL est un véritable serveur (ou gestionnaire) de base de données SQL multi-utilisateur et multi-threaded. SQL est le plus populaire langage de base de données dans le monde. **MySQL** est une configuration client/serveur ce qui consiste en un serveur démon mysql, différents programmes clients et des bibliothèques. SQL est un langage standardisé qui rend facile le stockage, la mise à jour et l'accès à l'information. Par exemple, vous pouvez utiliser le SQL pour récupérer des informations sur un produit ou stocker des informations client sur un site web. **MySQL** est suffisamment rapide et flexible pour gérer des historiques et des images.

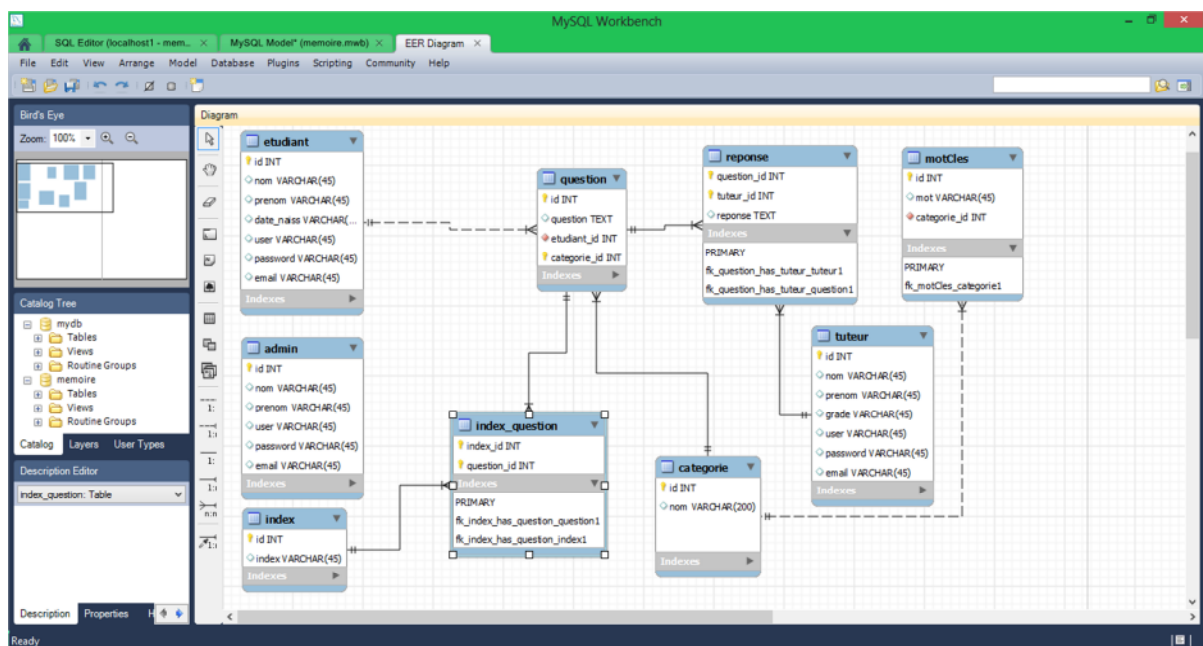


Figure III.6: interface "MySQL workbench 5.2"

III.4. Présentation de quelques interfaces :

III.4.1. Page d'accueil :

C'est la première page qui apparaîtra à tous utilisateurs de l'application.

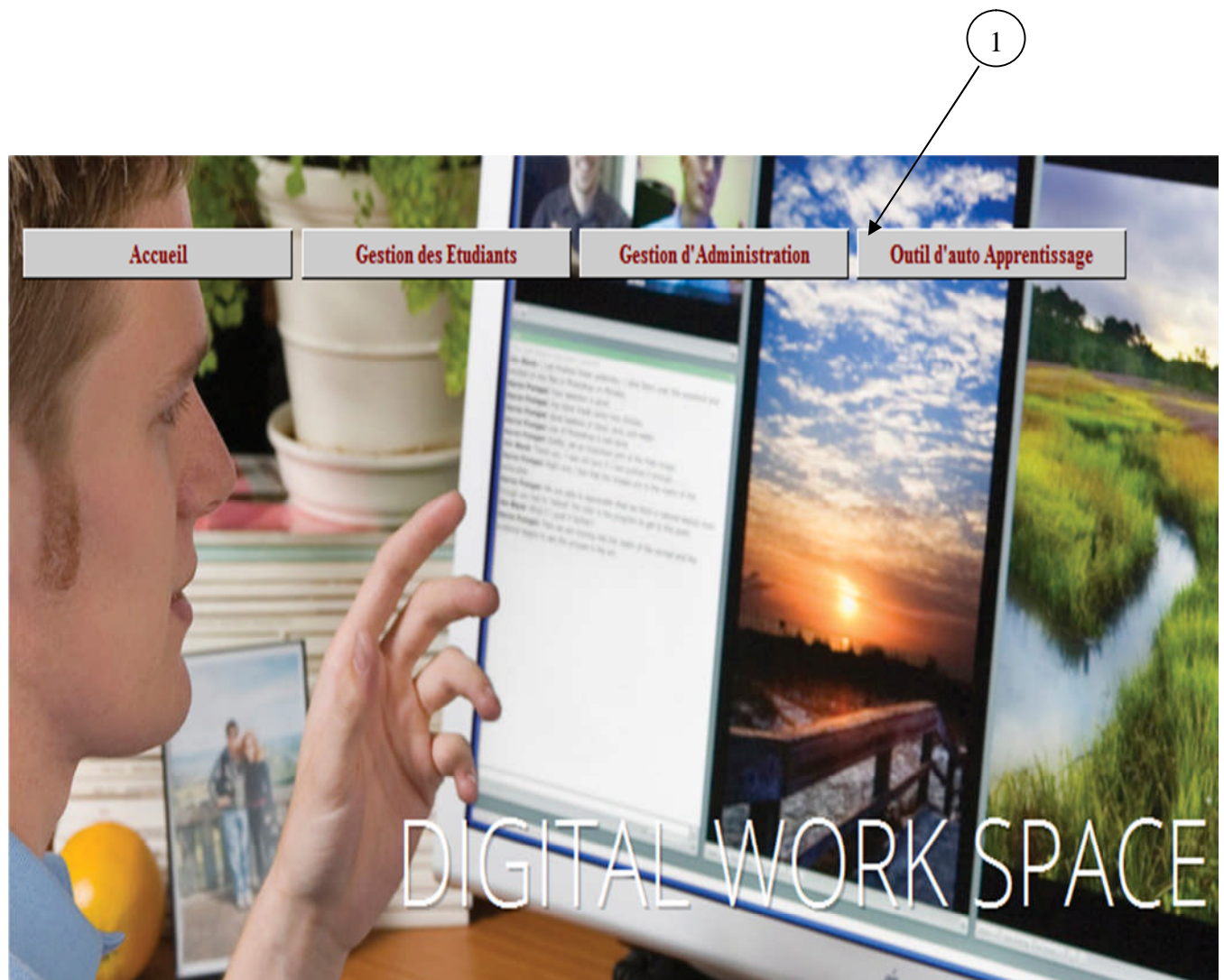


Figure III.7 : interface d'accueil

- ① : Donne accès au formulaire d'authentification des deux acteurs (Apprenant, Expert de domaine).

III.4.2. Page d'authentification et d'inscription apprenant :

The screenshot shows a web interface for 'Problem Based Learning...'. At the top, there are two buttons: 'inscription' (labeled with a circled '2') and 'Authentification' (labeled with a circled '1'). Below these buttons, the text reads: 'Vous devez faire l'inscription sur notre plate forme auto apprentissage:'. Underneath, there is a section titled 'Des informations personnelles' containing a registration form with fields for 'Nom:', 'Prénom:', 'Email:', 'Pseudo:', and 'Mot de passe:', each followed by an asterisk. A 'Valider' button is located below the registration fields. To the right, there is an authentication form with fields for 'Pseudo :' and 'Mot de pass :', followed by a 'Connexion' button.

Figure III.8 : Page d'authentification et d'inscription apprenant

- ① : Si l'apprenant déjà inscrit, il doit juste saisir son pseudo et le mot de passe, un clic sur le bouton « Connexion » le mène vers l'espace apprenant.
- ② : Si l'apprenant ne possède pas de compte dans la plateforme, il pourra faire une inscription en remplissant le formulaire d'inscription.

III.4.3. Espace Apprenant :

Dans son espace, l'apprenant aura les compétences dont sera évalué ; et les compétences qui seront affichées sont celle qui n'ont pas des pré-requis.



Figure III.9 : Espace Apprenant

1 : Le Plan guide adaptatif va générer les compétences qui n'ont pas des pré-requis.

2 : Le bouton de déconnexion.

The screenshot shows a web application titled "Problem Based Learning...". The main content area is titled "Mon espace de travail" and contains a section for "Compétence composée: Normalisation de Relations(Formes Normales)". Below this is the "Énoncé situation problème" which reads: "On considère la relation $R(A,B,C,D,E,F)$ et l'ensemble (S) formé par les dépendances fonctionnelles $AB \rightarrow C$, $BC \rightarrow AD$, $D \rightarrow E$ et $CF \rightarrow B$. Calculer $\{A,B\}^+$
1. Déterminer à l'aide de cette clôture si $AB \rightarrow D$ découle de S .
2. Même question pour $D \rightarrow A$." Below the text is a large "champ de saisie" (input field) with an "Enregister" button. To the right is a "SCAFFOLDING" panel with "CONSIGNES" and a "Revoir les cours" link. At the bottom right is a message box with "Ecrivez nous Geek", "votre message ici", and "envoyer" and "effacer" buttons. A "Bien venu Geek" message and a "PLAN GUIDE ADAPTATIF" menu are on the left. Three callouts are present: 1 points to the problem statement, 2 points to the scaffolding panel, and 3 points to the input field.

Figure III.10: Interface résoudre une situation de problème

- ① : L'énoncée situation problème sera afficher à chaque fois l'apprenant clique sur la compétence choisie.
- ② : Le scaffolding contient les consignes de la situation de problème, chaque situation de problème a ses propres consignes.
- ③ : Le champ de saisie de la solution de la situation de problème par l'apprenant.

III.4.4. Page d'authentification Expert du domaine :

The image shows a web interface for authentication. At the top, there is a dark blue banner with the text "Problem Based Learning..." in a large, glowing white font. Below this banner is a dark blue horizontal bar containing the text "Authentification Expert du Domaine" in white. The main body of the page has a light blue background. In the center, there is a login form consisting of two white input fields. The first field is labeled "Pseudo :" and the second is labeled "Mot de pass :". Below these fields is a button with the text "Connexion".

Figure III.11:interface d'authentification Expert du domaine

Cette page contient un formulaire d'authentification expert de domaine contenant :

- ✓ Une zone de saisie du pseudo.
- ✓ Une zone de saisie du mot de passe.

Une fois le pseudo et le mot de passe sont saisis, un clic sur le bouton « connexion» mène

L'expert du domaine dans son espace.

III.4.5. Espace expert du domaine :

The screenshot shows a web interface for updating general competencies. The main title is "mise à jour compétence générale". There are four tabs for different types of updates: "mise à jour compétence generale", "mise à jour compétence composée", "mise à jour compétence elementaire", and "mise à jour situation problème". A "Deconnexion" button is located in the top right corner. The interface is divided into two main sections. The left section, labeled "Ajouter une Compétence generale" (indicated by callout 2), includes an "Ajouter" button and a "Modifier/Supprimer une compétence" section. This section lists three forms: "Premiere Forme Normale 1FN", "Deuxieme Forme Normale 2FN", and "Troisieme Forme Normale 3FN", each with "Modifier" and "Supprimer" buttons. The right section, labeled "Ajout de compétence" (indicated by callout 1), is a form for adding a new competency. It includes fields for "Nom de la compétence générale:", "Seuil:", and "Prérequis:" (a dropdown menu showing "--Aucun prérequis--"), and an "Enregistrer" button.

Figure III.12: interface mise à jour compétence générale

1 : L'expert du domaine ajoute une compétence générale en précisant son pré requis.

2 : L'expert du domaine peut modifier et supprimer une compétence générale.



Figure III.13: interface mise à jour compétence composée

- 1 : L'expert du domaine ajoute une compétence composée en précisant la compétence générale ou elle appartient et son prérequis.
- 2 : L'expert du domaine peut modifier et supprimer une compétence composée.

2

1

mise à jour compétence élémentaire

[mise à jour compétence generale](#) [mise à jour compétence composée](#) [mise à jour compétence elementaire](#) [mise à jour situation problème](#) [Deconnexion](#)

Ajouter une Compétence Élémentaire

Modifier/Supprimer un compétence

- compétence elementaire

Votre sauvegarde a été bien effectuée

Ajout de compétence

Compétence composée:

Compétence élémentaire:

Seuil:

Prérequis de la compétence:

Figure III.14: interface mise à jour compétence élémentaire

① : L'expert du domaine ajoute une compétence élémentaire en précisant la compétence composée ou elle appartient et son pré requis.

② : L'expert du domaine peut modifier et supprimer une compétence élémentaire

The screenshot shows a web application interface for managing competencies. At the top, there is a blue header with the text "mise à jour compétence générale". Below the header, there is a navigation bar with four buttons: "mise à jour compétence generale", "mise à jour compétence compose", "mise à jour compétence elementaire", and "mise à jour situation problème", along with a "Deconnexion" button. The main content area is divided into two sections. The left section is titled "Ajouter une Compétence generale" and contains an "Ajouter" button. The right section is titled "Modifier/Supprimer une compétence" and contains a list of competencies with "Modifier" and "Supprimer" buttons for each. A detailed view of a competency is shown in a green-bordered box, titled "Modification de compétence". This view includes fields for "Nom de la compétence generale" (Theorie des dependances et norm), "Seuil" (10), and "Prérequis" (--Aucun prérequis--), along with a "Modifier" button.

Figure III.15: interface ajout des compétences générales sans pré-requis

Dans cette figure (**Figure III.15**) l'expert du domaine a ajouté trois compétences générales (Théories des dépendances et normalisation, les langages relationnels et le modèle entité association) en précisant qu'il n'a pas pré-requis entre ces compétences.

Dans l'espace apprenant, ces compétences générales seront toutes affichées, comme la montre la figure suivante :

The screenshot shows a web application interface with a dark blue header containing the text "A New Way To Learn...". Below the header, there is a light blue background. On the left side, there is a navigation menu with a blue oval button labeled "Bien venu Geek". Below this, there is a section titled "PLAN GUIDE ADAPTATIF" with a list of items: "Theorie des dependances et normalisation", "Compéteces Composées", "Les langages relationnels", "Compéteces Composées", "le Modele Entite Association", and "Compéteces Composées". A black arrow points to the first item in the list. In the center of the page, there is a red text message: "Bien venu a votre espace". On the right side, there is a blue button labeled "Deconnexion", a text input field with the placeholder "Ecrivez nous Geek" and "votre message ici", and two buttons labeled "envoyer" and "effacer".

Figure III.16 : Espace Apprenant affichage des compétences sans pré-requis



Figure III.17: interface modification une compétence générale avec pré-requis

Dans cette figure (**Figure III.17**) l'expert du domaine a modifié une compétence générale (les langages relationnels) en précisant qu'elle a de pré-requis avec la compétence (Théories des dépendances et normalisation)

Dans l'espace apprenant, la compétence générale (les langages relationnels), elle ne sera pas affichée dans le plan guide de l'espace apprenant, comme le montre la figure suivante :

A New Way To Learn...

Bien venu Geek

Deconnexion

Ecrivez nous Geek

votre message ici

envoyer effacer

Bien venu a votre espace

Figure III.18 : Espace Apprenant non affichage des compétences qui ont des prérequis

mise à jour situation de problème

[mise à jour compétence generale](#) [mise à jour compétence composée](#) [mise à jour compétence elementaire](#) [mise à jour situation problème](#) [Deconnexion](#)

Pour quel type de compétence : Choisir une compétence

Nom du problème:

Contenu :

Consigne :

Seuil :

Figure III.19: interface mise à jour situation de problème

L'expert du domaine peut ajouter plusieurs situations de problèmes pour chaque compétence, et chaque situation de problème contienne plusieurs consignes en cliquant sur le bouton « ajouter d'autres consignes ».

Conclusion :

Au niveau de ce chapitre nous avons présenté l'environnement de développement et l'implémentation de notre application. Nous avons aussi décrit quelques interfaces d'utilisation.

Conclusion Générale

Nous avons présenté à travers ce travail une étude relative à la conception et la réalisation d'un système d'aide à l'apprentissage des compétences disciplinaires . La discipline choisie dans notre cas c'est base de données.

Ce système permet essentiellement de :

- Produire des compétences dans l'ordre des pré-requis.
- Produire des situations problèmes de chaque compétence, en tenant compte des cas où les situations problèmes ont déjà été présentées et répondues.

Ce travail nous a permis :

- De comprendre la nécessité des approches par compétences dans les systèmes d'apprentissages.
- D'approfondir nos connaissances en matières programmation JAVA, SQL
- S'adapter à l'environnement de développement Netbeans.

Notre travail quoiqu'il présente diverses fonctionnalités, sera toujours perfectible pour une meilleure d'acquisition de savoir-faire, et de le faire intégrer dans les systèmes hypermédias adaptatifs éducatifs. Et développer d'une façon détaillée le module scaffolding et l'évaluation.

BIBLIOGRAPHIE

- [Zoua,2012]** :Zoualfakar JAMMOUL L'APPRENTISSAGE PAR LES COMPETENCES LE RÔLE DE L'EVALUATION AU COURS DE L'APPRENTISSAGE PAR COMPÉTENCE, Manuscrit auteur, publié dans "LES COMPETENCES PROFESSIONNELLE, France (2012)"
- [Berk, 2012]** :Tassadit Berkane hypermedia adaptatif éducatif : interface adaptative et gestion des profils apprenants, 15 Mars 2012
- [Berk &boua]** :Tassadit Berkane & Farida Bouarab-Dahmani, Modélisation ontologique des Compétences pour un hypermédia adaptatif éducatif,Colloque Doctoral International de l'éducation et de la formation,Nantes – 28, 29 novembre 2013
- [APP ,2011]** : L'approche par compétences Un levier de changement des pratiques en santé publique au Québec, INSTITUT NATIONAL DE SANTÉ PUBLIQUE DU QUÉBEC, Février 2011.
- [GUILBERT, L. 2000]** : Introduction à l'analyse qualitative, recueil de notes de cours.
- [SAVOIE-ZAJC, L. ET KARSENTI, T. (2000)]** : Introduction à la recherche en éducation, Éditions du CRP, Sherbrooke.
- [MARQUIS, C. ET SAUVÉ, L. (1995)]** : La sélection des médias. Théorie et pratique. Télé-université.
- [BISSONNETTE, S. GAUTHIER, C. ET RICHARD, M. (2005)]** : Proposition d'un modèle d'ingénierie pédagogique basé sur les recherches en efficacité de l'enseignement, Centre franco-ontarien de ressources pédagogiques, Ottawa.
- [BASQUE, J. (1999)]** : L'influence des théories de l'apprentissage sur le design pédagogique, Télé-université et École de technologie supérieure, notes de cours.
- [BRIEN, R. (1997)]** : Sciences cognitives et formation, Presses de l'Université du Québec, 3e édition.
- [Ben, 93]**: D.R.Benyon "Adaptive systems: A solution to Usability Problems", Journal of user Modeling and user adapted interaction, Kluwer, 3(1) pp. 1-22
- [Legendre, 2001]** : Sens et portée de la notion de compétence dans le nouveau programme de formation, Université de Montréal.
- [Lebrun, 2007]** : Théories et méthodes pédagogiques pour enseigner et apprendre.

[Bru, 96]: P. Brusilovsky, E. Schwarz, Gerhard Weber. "ELM--ART: An intelligent tutoring system on World Wide Web", 1996.

[Vas, 97]:J. Vassileva . "Practical architecture for user-modeling in a hypermedia based information system" , 1997.

[Duf, 98] : A.Dufresne "Manuel Excel. Adaptive Hypertext and Hypermedia" Discussion List. 1998.

[Nel, 70]: Nelson, T. "A file structure for the complex, the changing and the indeterminate" 1970.

[Scallon, 2005] : L'évaluation des apprentissages dans une approche par compétences, Éditions du Renouveau Pédagogique Inc.

[Tardif, 2006] : L'évaluation des compétences. Documenter le parcours de développement, Chenelière Éducation, Montréal.