

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

Conception et développement d'application de gestion d'emploi du temps

Réalisé par :

- AYAD Soriya
- BENMANSOUR Fatima Zohra

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

- *Mr : Smahi I. (Encadreur)*
- *Mr : Chouiti S. (Examineur)*
- *Mme : Iles N. (Examinatrice)*

Résumé

Ce modeste travail s'intéresse à la conception et la réalisation d'un système de gestion d'emploi du temps pour l'université, les objectifs principaux de ce système est de faciliter l'établissement des emplois du temps au niveau des départements, permet la création et la modification des emplois du temps ainsi la gestion de quelque conflit qui peuvent exister (conflit d'enseignant, salle).

ملخص

يركز هذا العمل المتواضع على تصميم وتنفيذ نظام إدارة الجدول الزمني للجامعة، والأهداف الرئيسية لهذا النظام هو تسهيل إنشاء جداول زمنية على مستوى الإدارات، ويسمح إنشاء وتعديل الجداول الزمنية وإدارة أي النزاعات التي قد تكون موجودة (سواء الأساتذة أو غرفة التدريس).

Resume

This modest work focuses on the design and implementation of a management system schedule for the university, the main objectives of this system is to facilitate the establishment of timetables at the departmental level, allows the creation and modification of schedules and management of any conflict that may exist (conflict of teacher or room).

Sommaire

Introduction générale.....	6
CHAPITRE 1	
I. Introduction.....	1
II. Système d'Information (SI)	1
II. 1.Définition.....	1
II. 2.Les cycles d'un système d'information	2
III. Génie logiciel(GL)	2
III .1.Définition.....	2
III.2.Les outils du GL.....	3
IV. Interface homme machine (IHM).....	3
IV.1.Définition	3
IV.2.Présentation de l'IHM :	3
V. Présentation du département d'informatique.....	4
V.1.Définition.....	4
V.2.Les services du département.....	4
VI. Présentation de notre thème	5
VI.1.Emploi du temps	5
VI.2.Type de plannings dans le domaine de la pédagogie	5
VI.3.Gestion de conflit	6
VII. Objectifs du projet	7
VII.1.La problématique	7
VII.2.Solutions Adopté :	8
VIII. Conclusion.....	8
CHAPITRE 2	
I. Introduction.....	9
II. Processus de développement logiciel	9
II.1.Processus Unifié	9
III. Spécifications des besoins	10
III.1.Langage UML (unified modeling language).....	10
III.1.1.Définition.....	10
III.1.2.Les outils du langage	11
III.2.La méthode « MERISE »	11
III.2.1.Définition.....	11

IV. Conception	12
IV.1.Expression du besoin.....	12
IV.1.1.Les ressources	12
IV.1.2.Les concepts généraux	12
IV.2.Les diagrammes des cas d'utilisation.....	13
IV.3.Spécifications des cas d'utilisations.....	14
IV.3.1.Cas d'utilisation « Gérer les ressources »	14
IV.3.2.Cas d'utilisation « Affectation des ressources »	15
IV.3.3.Cas d'utilisation « Consulter l'emploi du temps »	15
V. Analyse.....	16
V.1.Les diagrammes de séquence	16
V.1.1.Diagramme de séquence -Authentification :	16
V.1.2.Diagramme de séquence-Affecter un enseignement à un créneau horaire	18
V.1.3.Diagramme de séquence-Création d'un enseignement.....	19
V.1.3.Diagramme de séquence-Affecter un enseignement à un enseignant.....	20
V.2.Le Diagramme de classe.....	20
V.2.1.Définitions des classes.....	21
VI. Deuxième partie d'analyse.....	22
VI.1.Model Conceptuel de données(MCD).....	22
VI.2.Model logique de données.....	23
VII. Conclusion	23
CHAPITRE 3	
I. Introduction.....	24
II. Environnement de travail :.....	24
II.1.Solution Matériel :	24
II.2.Solution Logiciel :	24
II.3.Power AMC	25
II.3.1.Définition.....	25
II.3.2.Les Outils.....	25
II.4.Oracle Database	25
II.4.1.Définition de l'environnement.....	25
II.4.2.Définition des outils.....	26
II.5.Delphi :	26
III. Description de l'application	28

IV. Conclusion :	34
Conclusion Générale	35
Référencement bibliographie :.....	36
Résumé.....	2

Introduction générale

Il ne fait désormais plus aucun doute que l'informatique représente la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité en ce siècle passé.

L'ordinateur est un système de traitement automatique des informations qui permettent d'améliorer les performances humaines pour la manipulation rapide de grande information.

De cette constatation, nous nous sommes intéressés, dans ce travail, au développement d'une application facilitant l'établissement des emplois du temps au sein de la Faculté des sciences de l'Université de Tlemcen et plus précisément le département d'informatique.

Pour cela notre rapport est structuré en trois chapitres :

- ✓ Le premier chapitre présente le contexte général de notre projet. Il donne une idée sur notre problématique, nos motivations et nos objectifs et situe le cadre de notre travail.
- ✓ L'analyse des besoins qui est vient après l'étape de conception. Où nous avons détaillé les défaillances du système existant, ceci nous permet de proposer des solutions adéquates et les modéliser. Donc cette étude fait l'objet de notre deuxième chapitre.
- ✓ Le troisième chapitre illustre la réalisation de l'application et se termine par une partie de tests d'exécutions de notre application
- ✓ .Enfin, une conclusion synthétise notre travail et présente les perspectives envisagées.

CHAPITRE 1

I. Introduction

Une bonne gestion du temps permet d'être bien organisé, efficace et performant. L'objectif de ce projet est de mettre en pratique des concepts étudiés durant notre cursus universitaire de licence. Plus spécifiquement dans les modules des *Systemes d'information*, *Génie Logiciel*, *Objets Distribués* et *interface Homme machine (IHM)*.

Pour cela nous avons réalisé un système permettant la gestion d'emploi du temps, en prenant comme cas d'application les contraintes et les spécificités de notre département d'informatique.

Au-delà, comme première partie nous allons présenter le contexte général de notre travail, en montrant le but du projet, sa problématique, la spécification de nos besoins, ainsi qu'une gestion des conflits.

II. Système d'Information (SI)

II. 1. Définition

Un système d'informatique suit une logique rigoureuse, sa cohérence interne implique une séquence obligée des différentes opérations autrement dit le système d'information selon « MOIGNE¹ » est l'ensemble des méthodes et des moyens recueillant, contrôlant mémorisant et distribuant les informations nécessaires à l'exercice de tout point de l'organisation des quatre fonctions d'un système d'information :

1. **Collecter** les informations provenant des autres éléments du système ou l'environnement extérieur du système (fonction dynamique)
2. **Mémoriser** les données manipulées par le système (fonction statique)
3. **Traiter** les données stockées (fonction dynamique)
4. **Transmettre** : les informations vers les autres composantes du système ainsi que vers l'environnement extérieur au système (fonction dynamique) [1].

¹ MOIGNE : (Jean-Louis Le Moigne) Spécialiste français de la systémique et l'épistémologie constructiviste.

II. 2. Les cycles d'un système d'information

Dans les systèmes d'information on énumère trois cycles différents pour un système d'information, à savoir le cycle de vie, le cycle d'abstraction et en fin le cycle de décision :

1. **Cycle de vie** : Il traduit le cheminement chronologique du système d'information depuis sa création (naissance) et son développement, jusqu'à son obsolescence et sa remise en cause (mort).
 - **Conception** : fournit une description fonctionnelle et technique détaillée du système.
 - **Réalisation** : élaboration des programmes afin de mettre en œuvre les solutions techniques précédemment retenues.
 - **Maintenance** : permettra de prolonger la vie du système d'information.
 - **Déclin** : lorsque le système d'information sera jugé obsolète, un nouveau cycle de vie recommencera.
2. **Cycle d'abstraction** : les méthodes de conception et d'analyse d'un SI s'appuient fortement sur ce cycle, ça permet de traduire les différents degrés d'abstraction du SI, en ignorant les détails d'implantation. Les niveaux conceptuel, logique et physique représentent les trois niveaux dans ce cycle.
3. **Cycle de décision** : traduit l'ensemble des mécanismes de décision et de choix à prendre lors du développement du SI. Ce cycle est très important dans l'étude des systèmes d'information.[2]

III. Génie logiciel(GL)

III .1.Définition

Le génie logiciel (anglais *software engineering*) est une science de génie industriel qui étudie les méthodes de travail et les bonnes pratiques des ingénieurs qui développent des logiciels .Le génie logiciel s'intéresse en particulier aux procédures systématiques qui permettent d'arriver à ce que des logiciels de grande taille correspondent aux attentes du client, soient fiables, aient un coût d'entretien réduit et de bonnes performances tout en respectant les délais et les coûts de construction[3].

III.2. Les outils du GL

Le génie logiciel constitue un véritable enjeu stratégique pour le développement des applications informatiques, en particulier les évolutions stratégiques des ateliers de GL qui concernent :

- Les outils de développement (conception, production, fiabilité, maintenance).
- Les outils d'analyse et d'expertise.
- Les outils de gestion.
- Les outils de création et des de gestion de base de données.
- Les outils de modélisation, de conception et d'évolution des architectures.

Ces outils portent sur une partie ou sur l'ensemble du cycle de vie du logiciel.

IV. Interface homme machine (IHM)

IV.1. Définition

L'Interface Homme Machine représente l'élément clé dans l'utilisation de tout système informatique. Les interfaces de notre système sont conçues de manière à être simples, naturelles, compréhensible et d'utilisation faciles.

IV.2. Présentation de l'IHM :

Pour interagir avec la machine nous avons besoin d'intermédiaires pour cela les IHM jouent ce rôle.

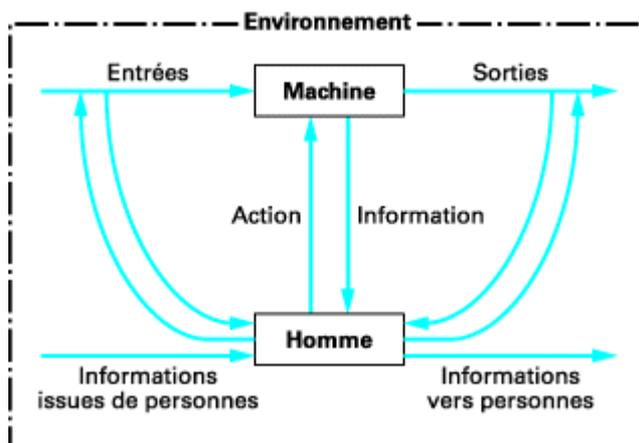


Figure I.1 : L'environnement de l'IHM

V. Présentation du département d'informatique

V.1.Définition

Le département d'informatique de l'université « Abou Bakr Belkaid » avait assuré depuis sa création en 1998 jusqu'à 2005, la formation du système classique des deux cycles de la graduation à savoir le cycle court en vue de l'obtention du diplôme de DEUA² en informatique et le cycle long en vue de l'obtention du diplôme d'ingénieur d'état en informatique (options : système D'information avancé et informatique industrielle). A partir de l'année universitaire (2005/2006) le département d'informatique a commencé à prendre en charge la formation du système L.M.D (Licence, Master et Doctorat).

V.2.Les services du département

Le département d'informatique est l'un des départements géré par la faculté des sciences sous la direction du Doyen, et par un chef de département, assisté par d'autres responsables chacun d'eux avec ses propres fonctions par rapport à son service.

On trouve au sein de ce département deux services:

1. Le service d'enseignement et de la scolarité sous la direction d'un chef adjoint à la pédagogie. Ce service assure la gestion des études et toutes les tâches concernant les enseignants et les étudiants du département. Les enseignants sont classés en fonction de leur grade. On distingue :

Les professeurs, les maîtres de conférences, les maîtres assistants et les chargés de cours, le corps des assistants, le jury de délibération, le promoteur et le conseil pédagogique.

2. Le service de la recherche scientifique sous la direction d'un chef de département adjoint à la recherche scientifique. Il gère les enseignants chercheurs et les étudiants en post graduation.

Les enseignants chercheurs organisent des réunions de travail pour exposer et étudier des thèmes de recherche, suivent les nouvelles technologies et animent des conférences pour l'information des étudiants.

Les étudiants chercheurs traitent les thèmes de recherches avec une approche symétrique et font évoluer les techniques.

² Diplôme d'Etudes Universitaire Appliquées

Les activités reliées au suivi pédagogique dans le département de l'informatique sont :

Convoquer les enseignants.

Etablir l'emploi du temps.

Organiser le CP.

Organiser les examens.

Organiser les soutenances.

VI. Présentation de notre thème

Comme déjà introduit auparavant, nous nous sommes intéressés à la problématique d'établissement des emplois du temps des différentes promotions de notre département.

Mais avant, nous présentons une idée générale sur le « concept » Emploi du temps.

VI.1. Emploi du temps

Qu'est-ce que le temps et sa gestion ?

- **Le temps** : est une ressource, un facteur de production disponible en quantité limitée, il permet de produire des biens et services, c'est une mesure de performance de toute organisation (administration publique ou entreprise).
- **La gestion** : administration, organisation et contrôle administratif d'une entreprise privée ou publique, organisation et actualisation de quelque chose (gestion d'emploi du temps).
- **Emploi du temps** : Fiche ou tableau donnant un programme d'activités (planning).
- **Gestion d'emploi du temps** : Définir des objectifs et des propriétés et mettre en œuvre un plan d'actions personnel de gestion du temps.

VI.2. Type de plannings dans le domaine de la pédagogie

La confection d'emploi du temps dans les établissements scolaires est un travail très important, difficile à réaliser les solutions fournies, nécessite d'être capable de s'adapter aux changements dynamiques de l'environnement en tenant compte de la diversité des contraintes telles que l'interdépendance des programmes d'enseignement, la multitude des matières étudiées et les contraintes sur ces matières (Cours, TD, TP), la durée des

cours, les contraintes de disponibilité des enseignants, la disponibilité limitée des salles. La confection de plannings d'horaires est donc une tâche très difficile et sa solution manuelle peut exiger beaucoup d'effort ce qui a attiré énormément l'attention de la communauté scientifique. Comme notre travail se rapporte au problème de résolution d'emploi du temps de notre département d'informatique, une large variété d'approches et modèles ont été proposés pour traiter une variété de problèmes d'emploi du temps. Les premières activités d'emploi du temps ont été effectuées manuellement et un emploi du temps typique, une fois construit est resté statique avec seulement quelques changements nécessaires. Cependant la nature des enseignements a changé considérablement au cours des années et ainsi les exigences en matière de confection d'emploi du temps sont devenues beaucoup plus compliquées qu'ils ont eu l'habitude de l'être. Par conséquent le besoin de la génération automatisée d'emploi du temps augmente et ainsi le développement d'un système de génération d'emploi du temps qui produit des solutions valables est essentiel.

VI.3. Gestion de conflit

Un conflit se caractérise par le partage de ressources entre plusieurs séances au même moment. L'idée de base de notre travail est de concevoir un outil qui permette interactivement de placer des séances dans un planning en garantissant l'absence de conflit. Eviter les conflits revient alors à éviter les erreurs de saisie. Les acteurs qui apprécient cette méthode sont les responsables des salles puisqu'ils attribuent les salles aux séances une fois que celles-ci sont planifiées. Cette méthode est également intéressante pour la création des emplois du temps puisqu'elle limite à chaque instant l'espace des séances qu'on peut placer sur un créneau donné. Dans ce cas, cette méthode seule ne suffit pas. Nous avons constaté que, lorsqu'il souhaite placer une séance sur un créneau, l'acteur est dérouteré si la séance qu'il pense pouvoir placer est absente de la liste des séances possibles. Pour résoudre ce problème, il faut expliquer les causes de cette absence (Enseignant ou salle occupé, ou groupe ou l'un de ses descendants ou ascendants occupé). Une autre façon d'aborder ce problème est d'accepter les conflits et de les résoudre après coup. Le danger extrême de cette méthode est qu'on risque d'avoir beaucoup de conflits qu'il sera impossible de résoudre sans recommencer le travail au point de départ. La résolution d'un conflit peut se faire de différentes manières.

Nous avons exploité une résolution des conflits qui va être présentée dans notre application.

VII. Objectifs du projet

Chaque année, les responsables pédagogiques de la direction des études ont pour mission de concevoir les emplois du temps des différentes filières en essayant, au mieux, de satisfaire les différentes contraintes rencontrées :

- Contraintes relatives aux enseignants (agenda, disponibilité,...) et aux étudiants,
- Contraintes pédagogiques imposées par la progression des enseignements,
- Contraintes « physiques » liées aux ressources matérielles (les salles, les équipements, etc.)

Ainsi, l'idée de mettre en place un système de gestion des emplois du temps dont l'objectif principal est de réaliser une application qui devra notamment, permettre :

- La génération automatique des emplois du temps.
- La consultation et suivi des emplois du temps.
- L'établissement des statistiques relatives aux informations enregistrées.

VII.1.La problématique

Dans l'université, la gestion de l'emploi du temps est une tâche importante et évidente puisqu'elle s'agit de gérer le temps de différentes personnes possédant leur propre activité et ceci de manière à satisfaire chacune d'entre elles, de ce fait les grands établissements se trouvent confrontés à plusieurs difficultés dont :

- La grande masse d'informations.
- Les critères d'optimisation mal définis.
- La diversité des contraintes et leurs interactions.

De ces difficultés découlent de nombreux problèmes vu le nombre importants des données et comme nous avons expliqué brièvement dans la partie précédente, nous avons pour but la conception d'un système permettant la gestion d'un emploi du temps.

VII.2.Solutions Adopté :

Dans la présentation des solutions adoptée, nous allons manipuler des objets en utilisant le langage de programmation **Delphi 7** et établir la connexion à la base de données qui est géré par le système de gestion de base de données **Oracle**, Ceci est fait en suivant les étapes suivantes :

- Elaboration du cahier des charges.
- Conception de la base de données avec Merise en élaborant le modèle conceptuel et le modèle physique de données et avec UML pour développer l'utilisation.
- Implémentation de la base de données.
- Développement de notre application(Programmation).
- Validation de l'application par des scénarios de tests d'exécution.

VIII. Conclusion

Nous pouvons conclure que la planification des horaires présente des enjeux à la fois sur un plan économique et un plan social. Toutefois, sa complexité impose de s'appuyer sur une démarche scientifique pour apporter des réponses pragmatiques à une catégorie générale de problèmes.

Il s'agit donc de développer un outil de planification d'horaires, basés sur des Techniques efficaces d'optimisation de ressources qui permettent de construire des programmes de travail, respectant la réglementation du travail et garantissant une bonne couverture de charge tout en limitant les coûts.

CHAPITRE 2

I. Introduction

La gestion des emplois du temps est un problème difficile, notamment dans les universités et consomme de nombreuses ressources humaines et financière ainsi les différentes actions qui s'effectuent sur le système EDT (Emploi Du Temps).

Cette partie est relative à l'analyse et la conception de notre outil interactif d'aide à la gestion des emplois du temps, capable de prendre en compte une organisation hiérarchisée des données et de maintenir la cohérence des contraintes sur ces données. Pour mener bien à notre projet nous débuterons ce chapitre par la partie conception où nous allons identifier et construire les cas d'utilisations du système puis analyser chaque chemin(cas d'utilisation) en construisant les diagrammes de séquence et nous terminons par le diagramme de classes. Nous avons opté le processus UP (unified processus) comme démarche d'analyse.

II. Processus de développement logiciel

Un processus définit une séquence d'étapes, en partie ordonnées qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant [5]. Nous avons utilisé le processus unifié (UP) :

II.1.Processus Unifié

Le Processus Unifié (PU ou UP en anglais pour Unified Processus) est un processus de développement logiciel, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. Les principales caractéristiques du processus unifié sont :

- a) Conduit par les cas d'utilisations, ce qui veut dire que la conception est réalisée conformément aux attentes des utilisateurs du produit.
- b) Piloté par les risques, dans le cadre, où les différentes sources d'échec du produit doivent être écartées. Une source importante d'échec d'un produit logiciel est son inadéquation aux attentes de l'utilisateur.

La gestion du processus est organisée suivant les cartes phases suivantes :

Initialisation, élaboration, construction, transition.

c) Centré sur l'architecture, car il impose le respect des décisions d'architecture à chaque état d'instruction du modèle.

d) itératif et incrémentale, au sens que la réalisation du produit se fait en plusieurs itérations où chaque itération aboutit à livrer une partie de produit (un module), l'ajoute aux différentes parties déjà construit (incrémental) en gardant un produit homogène.

III. Spécifications des besoins

Pour atteindre nos objectifs dans des meilleures conditions, nous devons spécifier les besoins de notre application (gérer l'emploi du temps), pour cela notre travail va se réaliser sur deux étapes :

Utilisateurs Systèmes : On distinguera deux grands types d'utilisateurs du système :

Les administrateurs ayant le droit de créer d'autres utilisateurs et de spécifier leur droit d'accès.

Les utilisateurs ayant tout juste le droit d'utiliser le système sans avoir de privilège administrateur.

Et pour effectuer cette étape nous avons utilisé un langage de modélisation « UML » et une méthode de développement « merise » .

III.1.Langage UML (unified modeling language)



III.1.1.Définition

C'est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions lancé par *l'omg* (Object Management Groupe) dans but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets .Elle est utilisée pour spécifier un logiciel et/ou pour concevoir un logiciel.

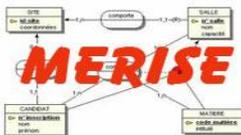
III.1.2. Les outils du langage

Le langage UML permet de définir et de visualiser un modèle, à l'aide de diagramme, en plus particulièrement nous avons utilisé les diagrammes suivants :

- des diagrammes de cas d'utilisation.
- Des diagrammes de séquences qui permettrons de présenté de manière générale les étapes importantes d'exécution de notre système.
- Un diagramme de classe de niveau analyse.

Ces diagrammes UML sont la représentation graphique, qui s'intéresse à un aspect précis du modèle, et chaque type de diagrammes possède une structure qui véhicule une sémantique précise et la combinaison des différents type de diagramme offrent une vue complète des aspects statiques et dynamiques d'un système [5].

III.2. La méthode « MERISE »



III.2.1. Définition

C'est une méthode qui permet la conception et le développement de système d'information à travers une base de données relationnelle permet la limitation de la redondance de l'information

III.2.2. Les techniques

La méthode vise à concevoir l'information de chaque domaine de l'entreprise en suivant un processus de modélisation à deux niveaux

- **conceptuel** : son objectif est de répondre à la question quoi, quoi faire et quelles données et de comprendre le problème, et le modèle proposés pour merise au niveau conceptuel est le model conceptuel de données(MCD).
- **Logique** : à ce niveau tous choix organisationnels sont réalisés afin de déterminer, qui fera quoi, où et quand ??, et imposer ce choix. Le model proposé pour merise c'est le modèle logique de données(MLD) [1].

IV. Conception

Cette étape représente le départ de toute réflexion concernant la conception d'application informatique, elle présente une vue abstraite du système entier.

Cas d'utilisation

Les cas d'utilisations nous permettrons de définir d'une part le comportement du système en fonction de ces acteurs et d'autre part les données de base entrant en jeu lors des échanges clients / serveur et de la persistance des données. Chaque cas d'utilisation est accompagné d'une description textuelle permettant de définir de façon précise les différentes notions introduites.

IV.1.Expression du besoin

La gestion de l'emploi du temps met en relation 3 différentes ressources : les enseignants, les salles, les étudiants(les promotions).L'objectif est d'affecter des enseignements à un enseignant pour une promotion d'étudiants dans une l'emploi du temps : L'administrateur et Le responsable de formation.

- Le rôle de l'administrateur est de gérer les ressources. Il peut créer, modifier, et supprimer les formations d'enseignement, des enseignants, etc. Il peut aussi spécifier les dates de début et de fin d'un semestre.
- Le rôle du responsable de formation est le plus important dans la gestion d'emploi du temps c'est lui qui réalise l'affectation des ressources.

IV.1.1.Les ressources

- La notion formation est identique à celle que nous connaissons à la faculté.
- Un enseignement est similaire à un module, il concerne une matière particulière et peut être géré par plusieurs enseignant.
- Une promotion est l'ensemble des étudiants inscrits à une même formation.
- Un responsable de formation est considéré comme un enseignant.

Aussi d'autres ressources qui intervienne sur le système salle, jour...

IV.1.2.Les concepts généraux

Un semestre est une période d'une année dont laquelle sont dispensées les cours. Au cours d'une année il n'y a que 2 semestre.

Type de séance (Cours, Td, TP) sont des activités sur un créneau donné correspondant à un ensemble dispensé par un enseignant dans une salle pour une section (COURS) ou un groupe (TD, TP).

Un créneau définit par un jour et un horaire exemple : jeudi de 11h30 à 13h.

Section, Groupe décrit un ensemble d'utilisateurs distincts.

Horaire et jour c'est l'intervalle de date qui est définit par un jour et une heure de début et de fin.

Type jour et type horaire définit la périodicité d'un intervalle de date.

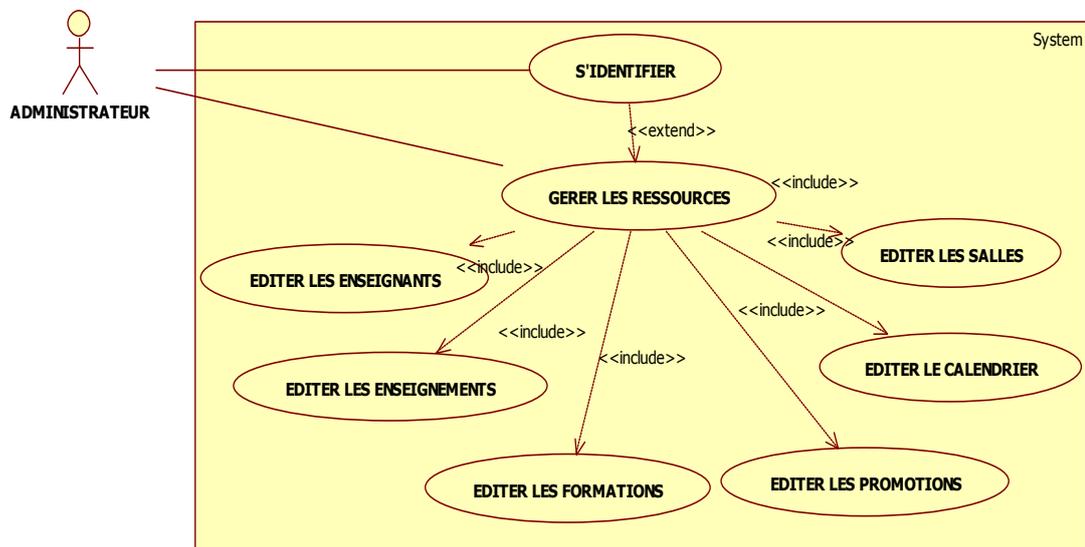
Calendrier ensemble de dates qui définit les périodes disponible pour les cours.

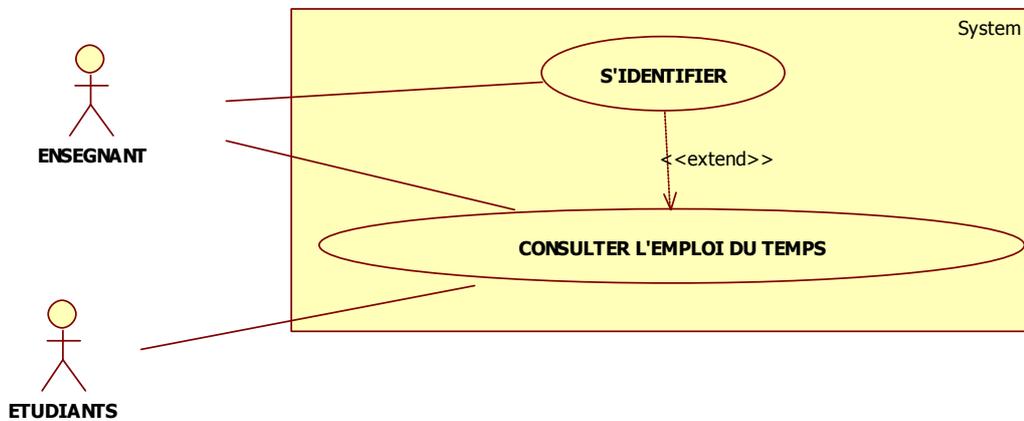
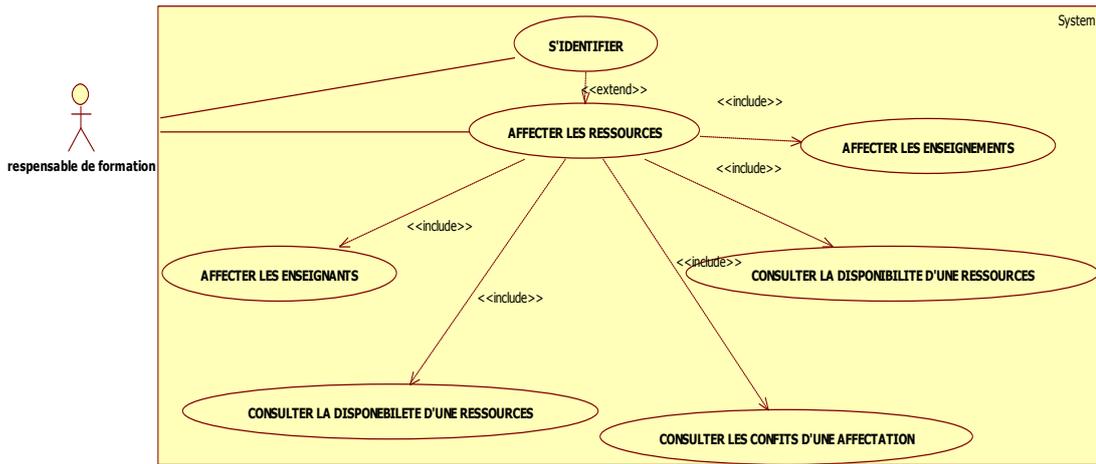
Emploi du temps regrouper les informations concernant tel jour tel horaire.

Salle lieu ou sont dispensés les enseignements.

Enseignant personne chargée de transmettre ses connaissance dans un domaine précis (enseignement).

IV.2. Les diagrammes des cas d'utilisation





IV.3.Spécifications des cas d'utilisations

IV.3.1.Cas d'utilisation « Gérer les ressources »

- ✚ Acteur : Administrateur
- ✚ Partie permanente et intérêt : l'administrateur : gère et alimente la base de données du système en introduisant les ressources nécessaire (dates, enseignant, salle) ainsi que la création, modification, ajout et suppression d'une ressource qu'il a préalablement choisi, cela afin d'avoir des informations en concordance avec la réalité.
- ✚ Pré_condition : l'administrateur possède ses identifiants.
- ✚ Poste_condition : les informations ont été enregistré.
- ✚ Scénario nominal :
 1. L'administrateur s'identifie au système avec ses identifiants.

2. Le système présente les informations actuellement présenté (présenter les ressources).
3. L'administrateur gère les ressources.
4. L'administrateur soumet les modifications au système.
5. Le système enregistre les modifications et signal le succès de l'opération.

✚ Extension :

- Condition : à tout moment
- Action: l'administrateur peut quitter à l'action en cours.
- Condition : l'identification échoue.
- Action : un message d'erreur apparait et retour en l

IV.3.2.Cas d'utilisation « Affectation des ressources »

✚ Acteur: Responsable de formation.

✚ Partie permanente et intérêt : le responsable de formation veut réaliser l'affectation des ressources.

✚ Pré_condition: le responsable de formation possède ses identifiants et des ressources qui existent.

✚ Poste_condition: les informations ont été enregistrées.

✚ Scénario nominale:

1. Le responsable de formation s'identifie au système avec ses identifiants.
2. Le système présente les informations actuellement présenté (présenter les ressources).
3. Le responsable de formation fait les affectations
4. Le système enregistre les modifications et signal le succès de l'opération

✚ Extension :

- Condition : à tout moment
- Action : le responsable de formation peut quitter à l'action en cours.
- Condition : l'identification échoue.
- Action : un message d'erreur apparait et retour en 1.

IV.3.3.Cas d'utilisation « Consulter l'emploi du temps »

✚ Acteur: enseignant.

✚ Partie permanente et intérêt : l'enseignant veut consulter l'emploi du temps

✚ Pré_condition : l'enseignant possède ses identifiants.

- ✚ Poste_condition : les informations ont été enregistrées.
- ✚ Scénario nominale :
 1. L'enseignant s'identifie au système avec ses identifiants.
 2. Le système présente les informations actuellement présente.
- ✚ Extension :
 - Condition : à tout moment
 - Action : l'enseignant peut quitter l'action en cours.
 - Condition : l'identification échoue.
 - Action : un message d'erreur apparaît et retour en 1

V. Analyse

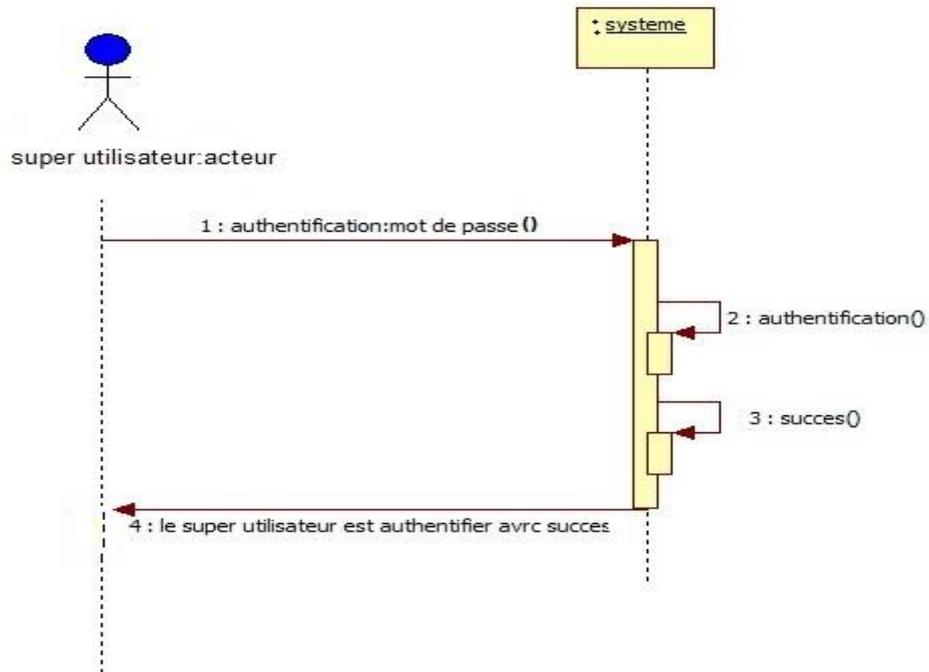
A partir des différents cas d'utilisation définis auparavant, nous avons créé différentes interfaces pour les services fournis par le serveur. Ces interfaces représentent chacune un ensemble de cas d'utilisation différents selon l'utilisateur qui demande l'accès aux services. Et afin d'illustrer certains fonctionnements du système, nous avons développé dans cette étape le modèle dynamique ainsi que le modèle statique où nous établirons des diagrammes de séquence et de classe pour visualiser le déroulement général des opérations.

V.1. Les diagrammes de séquence

Les diagrammes de séquence sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique, ils permettent de définir plus précisément que dans les cas d'utilisation le principe de fonctionnement de certaines phases d'application. De plus ils mettent en valeur certains choix de modélisation, nous en ferons la remarque lorsque cela sera nécessaire.

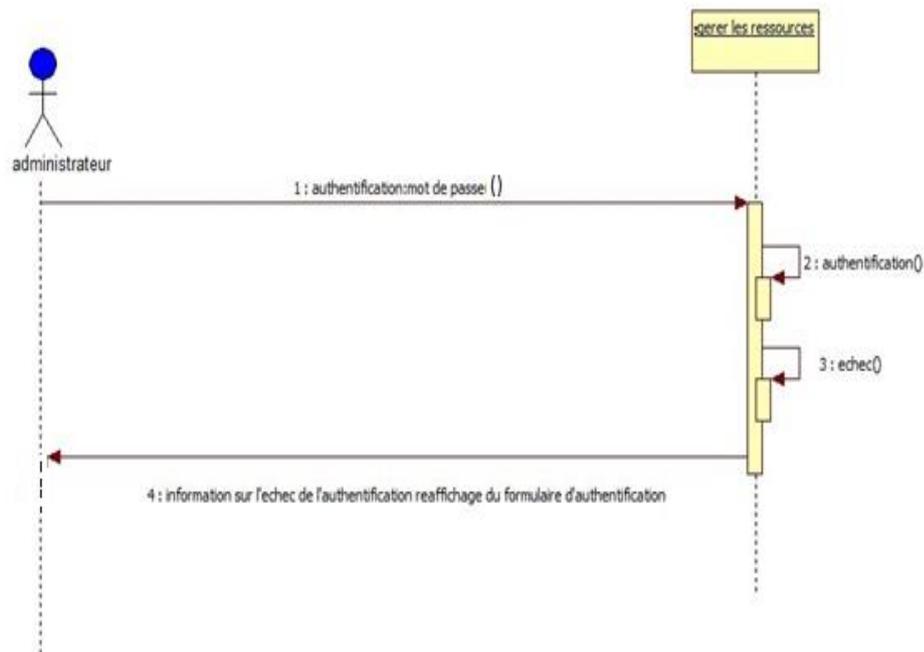
V.1.1. Diagramme de séquence -Authentification :

- Succès :



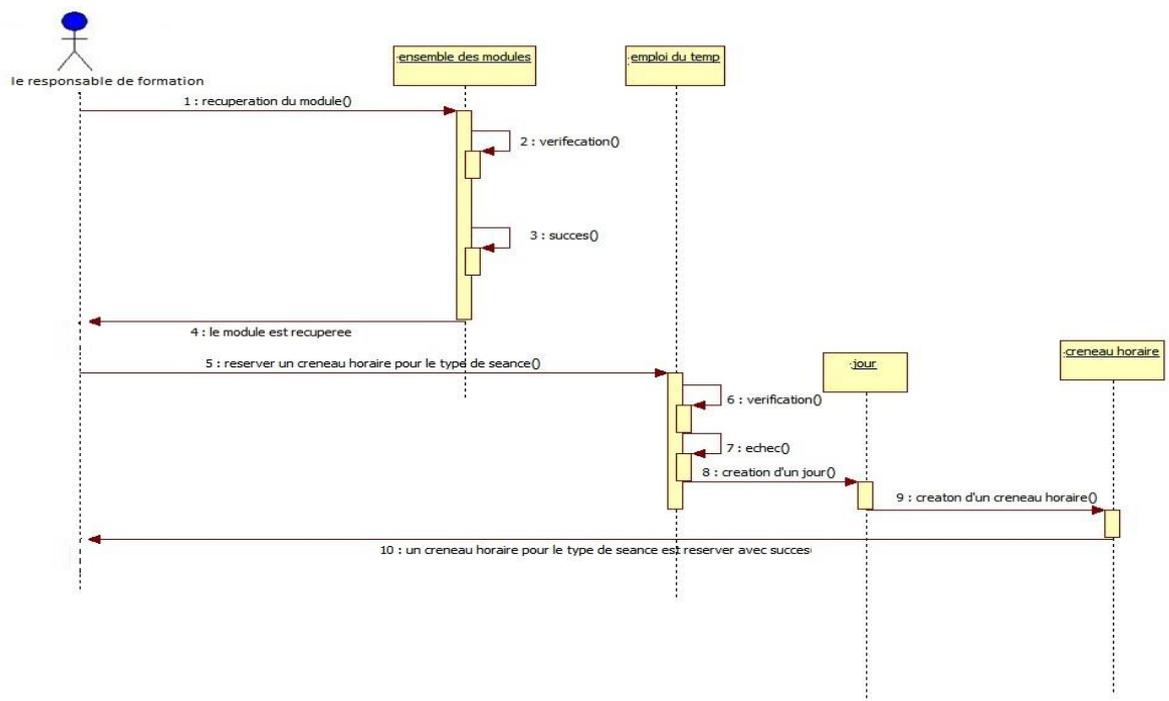
Ce diagramme est très simple à comprendre, il représente l'authentification d'un acteur par rapport au système. L'acteur peut être soit l'administrateur, soit le responsable de formation ou bien l'enseignant. La procédure est la même pour tous et utilise le système de login \ mot de passe.

- Echec :



Le diagramme de séquence ci-dessus montre une erreur lorsque l'administrateur souhaite s'identifier. Il permet notamment de mettre en avant que cette étape peut être infinie si l'erreur se répète. Une sécurité éviter/limiter les possibilités de piratage de compte pourrait être mettre une temporisation après un certain nombre d'échecs. Bien sur ce diagramme est valable pour le responsable de formation.

V.1.2. Diagramme de séquence-Affecter un enseignement à un créneau horaire



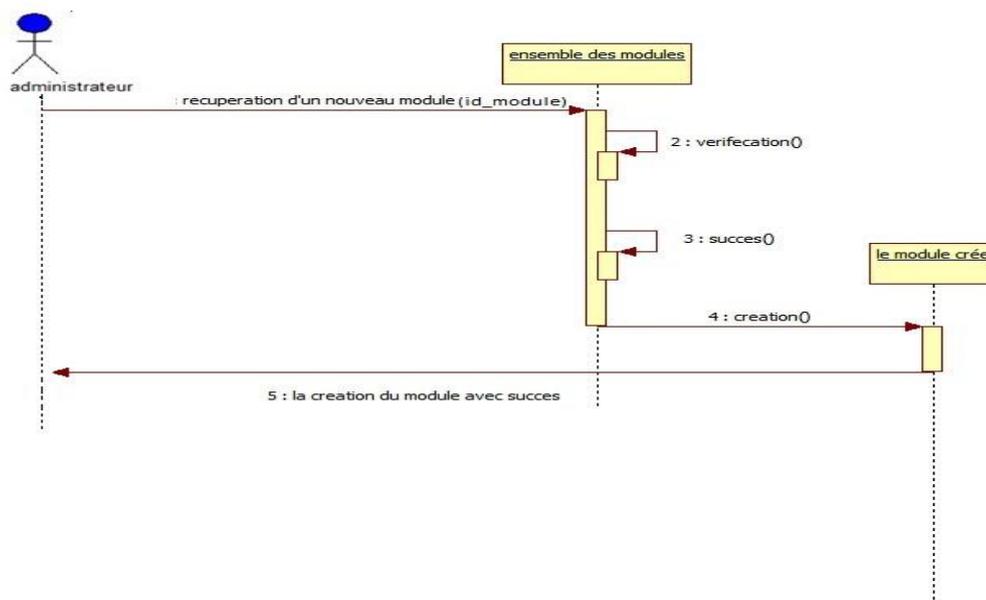
Dans ce diagramme de séquence le responsable de formation va prévoir une date pour un type de séance d'un enseignement, ce cas d'utilisation fait apparaitre la complexité du système que nous devons modéliser. Nous avons des données complexes et liées à manipuler :

- Un Enseignement

- Une Salle
- Un Enseignant
- Une date

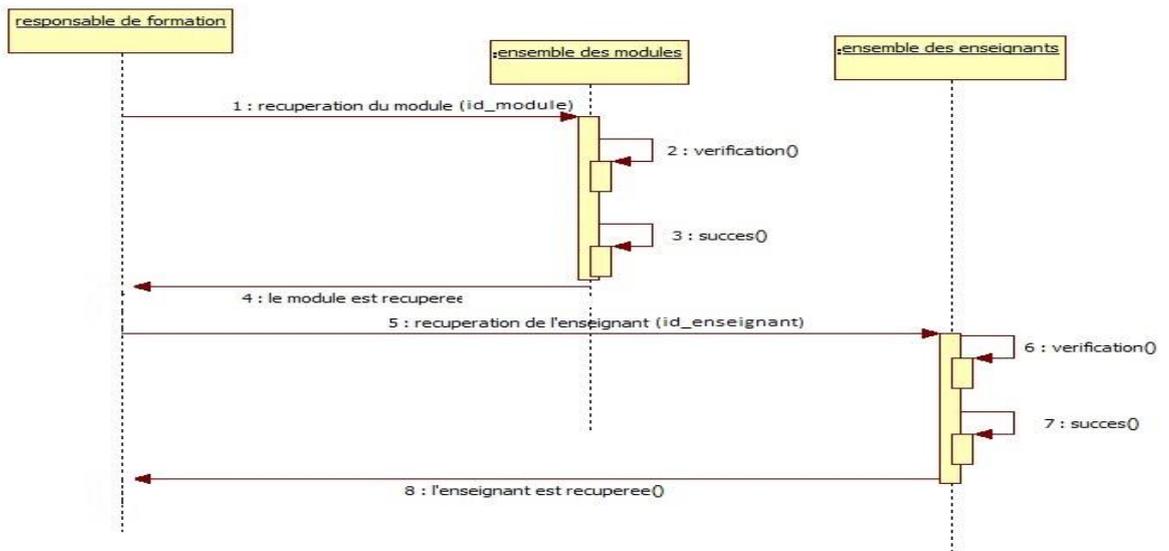
Nous avons choisi de ne pas traiter tout d'un coup, le responsable de formation va donc d'abord s'occuper de choisir un créneau horaire pour un enseignement. Le choix de salle et de l'intervenant interviendra après. Nous dissociions ainsi les contraintes matérielles (les ressources) des contraintes liées à l'enseignement, nous remarquons également ici que le jour est créé dynamiquement (s'il n'existe pas déjà) ainsi que le créneau horaire, l'échec montre que le jour n'existe pas encore.

V.1.3. Diagramme de séquence - Création d'un enseignement



L'administrateur souhaite créer un nouveau enseignement, tout d'abord le système vérifie si la matière n'existe pas déjà. Si tel est le cas la matière est créée.

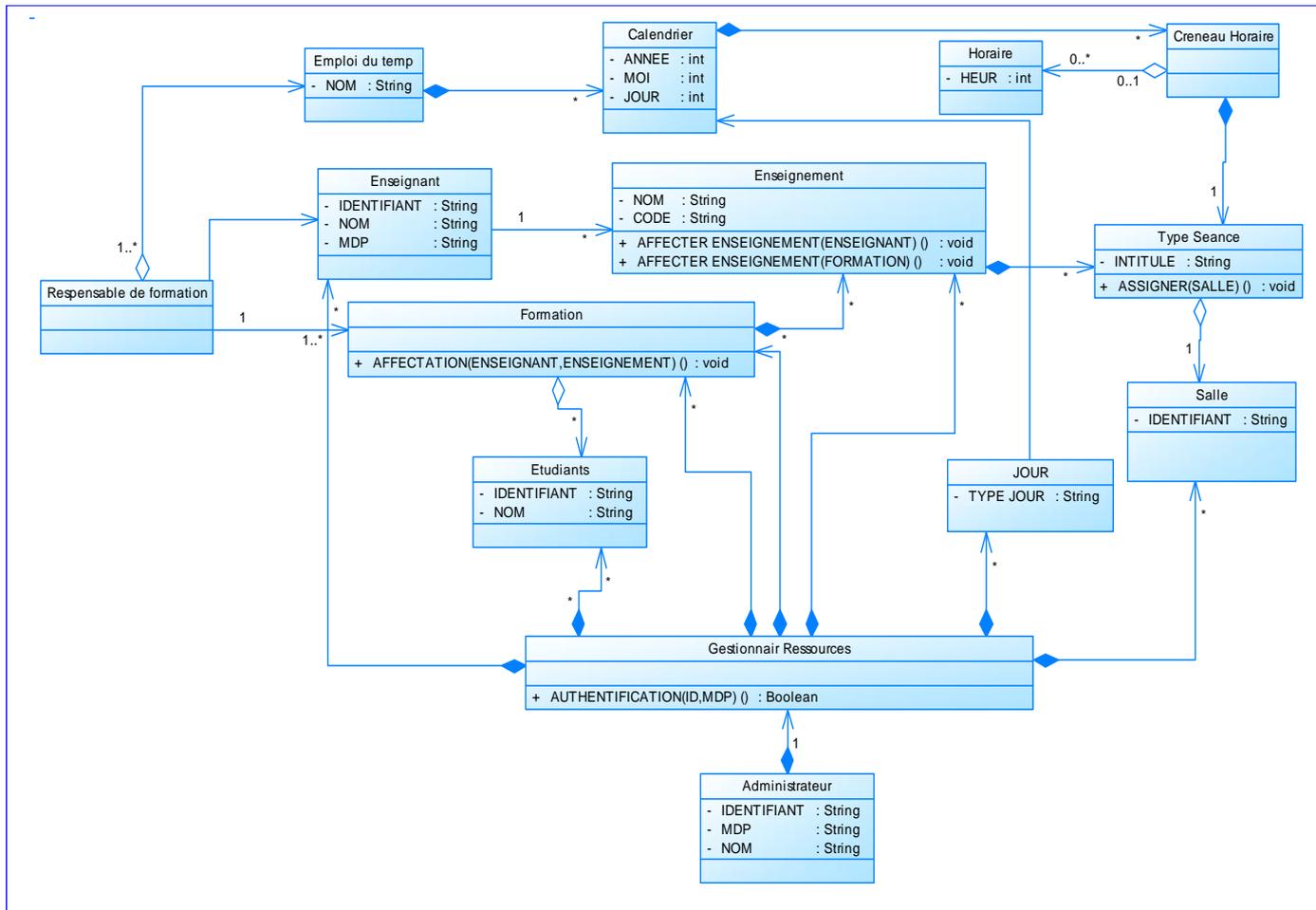
V.1.3. Diagramme de séquence-Affecter un enseignement à un enseignant



Le responsable de formation veut affecter un module pour un enseignant, il doit donc récupérer l'instance du module comme celle de l'enseignant pour les lier, l'enseignant étant lié au module il sera envisageable qu'il enseigne tous les type de séance du module, notre modélisation permet d'éviter ce problème, en effet plusieurs enseignant peuvent être lié a un module et en conséquent les type de séance pourront être reparti entre différent enseignants.

V.2. Le Diagramme de classe

Ce diagramme permet de représenter la structure statique d'une application, à savoir les éléments (classes et types), la structure interne des éléments et leurs relations les uns par rapport aux autres, le diagramme de classe que nous présenterons ici au niveau analyse c'est-à-dire que chaque classe ne correspondra pas forcément à un objet par la suite. En effet certaine classe pourront devenir des composants vu d'autre structures plus ou moins complexe. Chaque classe représente ici un concept important qu'il est essentiel de modéliser pour ensuite pouvoir réaliser l'application sans utiliser de solution miracle. Il met en relation toutes les informations exposées dans les diagrammes précédent et présente un condensé de toutes les réflexions faites dans ce chapitre.



V.2.1.Définitions des classes

- Un administrateur gère les ressources du système à savoir création, modification, suppression des concepts suivantes : salle, module (enseignement), calendrier, enseignant grâce à la classe gestionnaire des ressources, chacun de ces concepts est représenté par une classe.
- Le responsable de formation est aussi enseignant, cette notion se traduit par la relation d'héritage entre la classe enseignant et responsable de formation
- Nous autorisons le fait qu'un responsable de formation puisse gérer une ou plusieurs formations, ces pourquoi nous avons une multiplicité de 1..* entre la classe responsable et formation
- Pour chaque formation, le responsable choisi quelle enseignant va enseigner quelle module grâce à la méthode affecter enseignant de la classe formation
- Chaque type de séances est affecter a une salle, ce concept est représenté par la relation d'aggragation entre classe type de séances et salle, un type de séances est aussi placé dans l'emploi du temps(représenté par la classe emploi du temps)

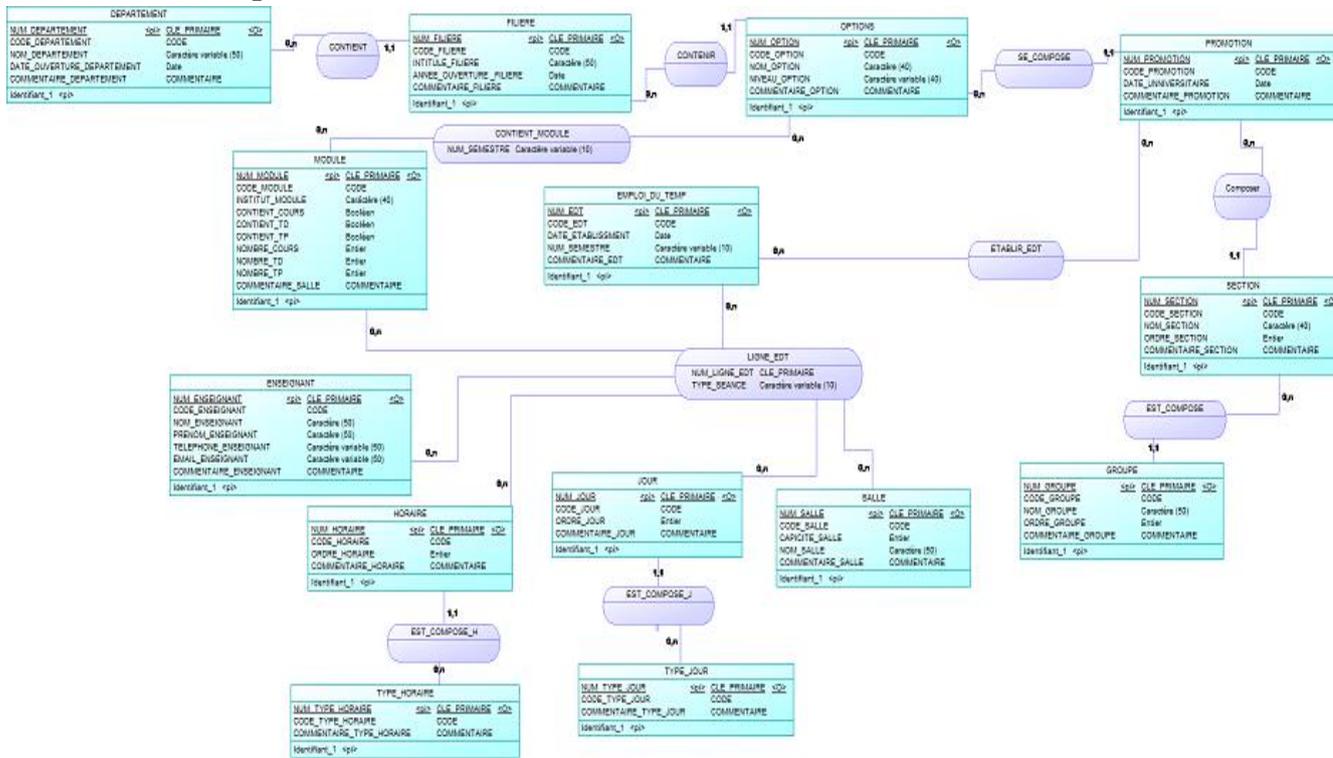
à un créneau horaire(représenté par la classe créneau horaire) et une date précise représenté dans la classe calendrier, chaque type de séances et dépend d'un module(enseignement).

VI. Deuxième partie d'analyse

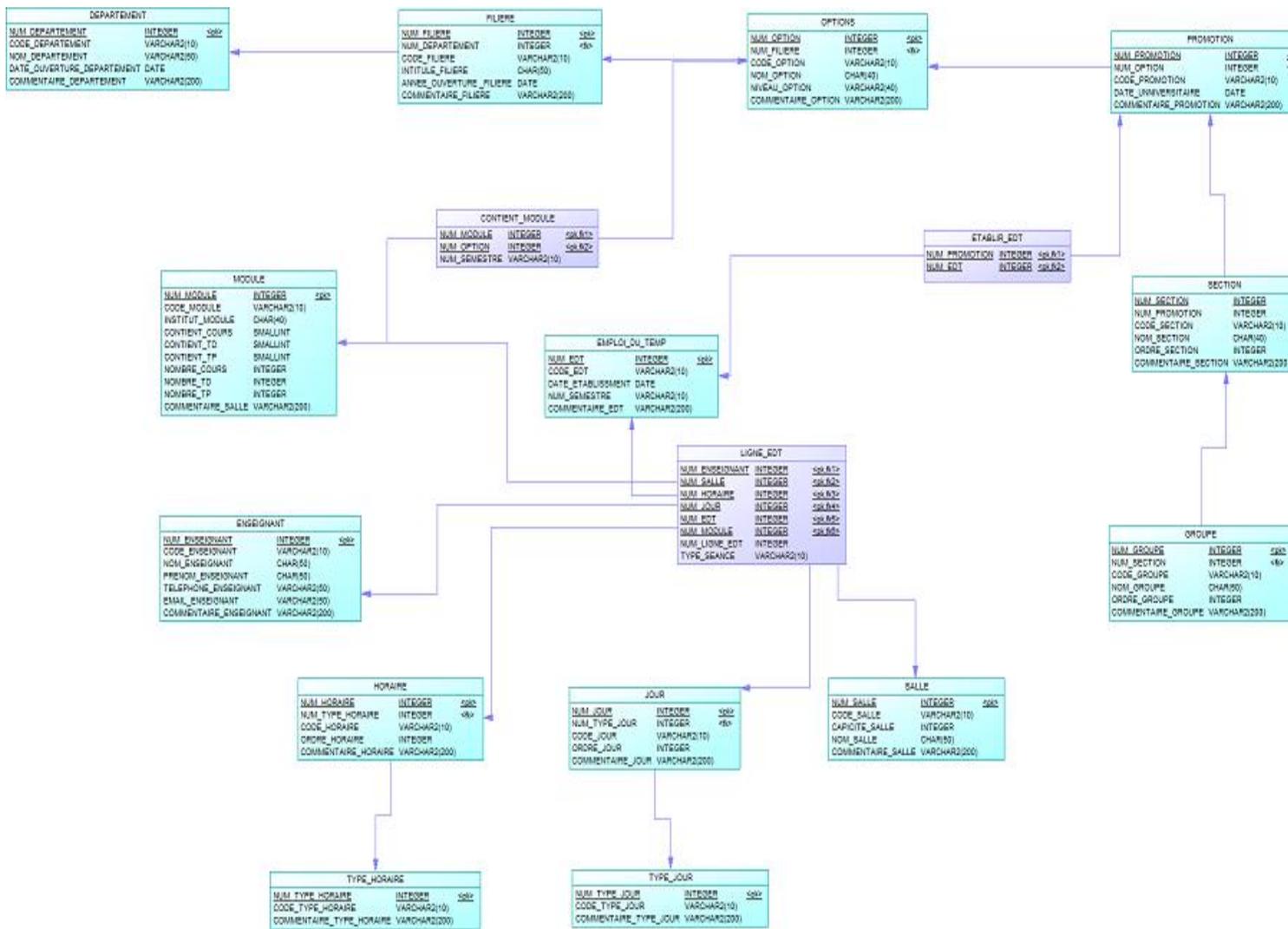
Après avoir étudié et analyser notre problème (gestion emploi du temps)

Nous avons pu créer à partir de diagramme de classe nos propres modèles conceptuels et logiques de données qui nous permettent à générer la base de données de notre application

VI.1. Model Conceptuel de données(MCD)



VI.2. Model logique de données



Pour réaliser cette étape on a utilisé Power AMC que nous le définir dans le chapitre suivant

VII. Conclusion

Dans ce chapitre, nous avons présenté la modélisation de notre système. Nous avons établie une modélisation de base caractérisant notre système. Dans ce chapitre, nous avons aussi, effectué une étude détaillée pour la mise en œuvre de notre système. La mise en place de ce dernier sera présentée dans le chapitre suivant.

CHAPITRE 3

I. Introduction

Après avoir finalisé l'étape de conception, nous passons dans ce chapitre à l'implémentation de notre application. Cette application a permis en premier lieu de répondre aux besoins des utilisateurs par la résolution des problèmes de la gestion de l'emploi du temps, qui est l'étape la plus importante dans ce travail. En deuxième lieu, elle a permis d'orienter le personnel vers une application unique. En effet l'application réalisée fournit un point d'accès unique pour les informations importantes du département. Plusieurs technologies ont été nécessaires pour la réalisation de notre projet, on citera donc le langage UML pour la modélisation statique et dynamique, SQL pour l'élaboration des requêtes d'interrogation de la base de donnée, et enfin le Delphi pascal pour l'écriture du code. Nous enchaînons par la présentation du travail réalisé et nous terminons par une partie tests et perspectives.

II. Environnement de travail :

Nous commençons tout d'abord par la définition de notre environnement matériel et logiciel.

II.1.Solution Matériel :

L'environnement matériel utilisé est le suivant :

Un microordinateur (système d'exploitation Windows XP).

II.2.Solution Logiciel :

Pour concevoir et implémenter notre application, nous avons utilisé les logiciels suivants :

- **Power AMC Designer 15.6** : pour la phase de conception de la base de données.
- **Oracle 10g** : pour implémenter la base de données.
- **Delphi 7**: fourni tous les outils nécessaires pour développer, tester, déboguer et déployer l'application.

II.3.Power AMC

II.3.1.Définition

Est un puissant outil de conception de base de données. Il vous offre tous les avantages d'une approche de conception à deux niveaux : vous conservez les modèles tant au niveau conceptuel et physique. Avec *PowerAMC* nous avons généré un script de création de base de données pour le *SGBD Access*, pour avoir implémenté des tables de base de données de l'application.

II.3.2.Les Outils

Le script généré par ce logiciel est en langage SQL qui est un langage normalisé servant à exploiter des bases de données relationnels .On distingue quatre partie dans ce langage :

- La partie *langage de manipulation des données* de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.
- la partie *langage de définition des données* permet de créer et de modifier l'organisation des données dans la base de données
- la partie *langage de contrôle de transaction* permet de commencer et de terminer des transactions.
- la partie *langage de contrôle des données* permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes.

II.4.Oracle Database

II.4.1.Définition de l'environnement

Est un système de gestion de base de données relationnel (SGBDR) qui depuis l'introduction du support du modèle objets dans sa version 8 peut être aussi qualifiée de système de gestion de base de données relationnels-objet (SGBDRO). Fourni par Oracle Corporation

Il est présenté sous forme de serveur web où on peut y accéder en connectant par un utilisateur système et un mot de passe.

Les étapes suivies pour générer la base de données :

- ❖ Création des tables.

- ❖ Création des index.
- ❖ Création des séquences.
- ❖ Création des déclencheurs (trigger).

II.4.2. Définition des outils

- **Index** : est un objet de base de données destiné à améliorer les performances des requêtes SQL, une structure de données redondante organisée de manière à accélérer certaines recherches.
- **Séquence** : Définir une séquence équivaut à définir une suite de nombres entiers. L'utilisation d'une séquence permet donc d'avoir à disposition une suite de valeurs. Ceci peut permettre de générer des clés uniques dans des tables et aussi avoir un compteur à titre informatif, que l'on incrémente quand on veut.
- **Trigger** : Un trigger est un programme qui se déclenche automatiquement suite à un événement (L'événement est une instruction du DML qui modifie la base de données. (INSERT, DELETE, UPDATE)).

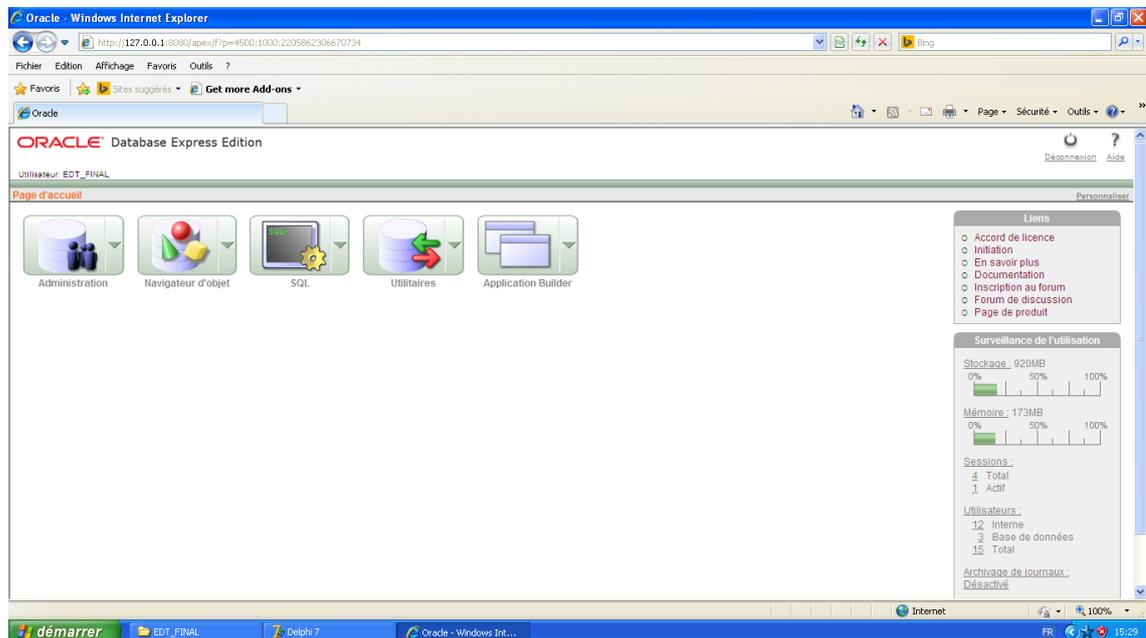


Figure III.1 : Oracle

II.5. Delphi :

Nous décrivons ici **Delphi** développé par la société **BORLAND** qui s'appuie sur une programmation en **PASCAL**. C'est un outil largement utilisé dans l'industrie, il fournit

à l'utilisateur des sous programmes tous faits permettant de visualiser à l'écran de telles fenêtres.

Après son lancement, Delphi se présente sous la forme de 4 fenêtres dans lesquelles on peut introduire les données et lire les résultats :

La première fenêtre occupe la partie supérieure de l'écran. Elle correspond à l'environnement de programmation proprement dit.

Cette fenêtre contient :

- la barre de titre.
- la barre de menu de Delphi.
- une zone « barre d'outils » (sur la gauche) .
- une zone contenant les divers composants regroupés par familles.

La seconde fenêtre se trouve par défaut à gauche de l'écran : c'est *l'inspecteur d'objets*. Il permet de

visualiser, pour chaque objet ou composant, les propriétés et les événements aux quels l'objet peut répondre.

La troisième fenêtre constitue la fiche principale de la future application Delphi. Il s'agit, au départ d'une fenêtre vide dans laquelle on placera les divers objets.

La dernière fenêtre, cachée sous la précédente constitue l'éditeur proprement dit, contenant le code source de l'application.

Et pour réussir notre application nous avons utilisé une importante bibliothèque de composants réutilisables et un ensemble d'outils de conception pour simplifier le prototype et réduisent la durée de développement et ces outils sont :

Au lancement du logiciel l'écran a l'aspect ci-dessus :

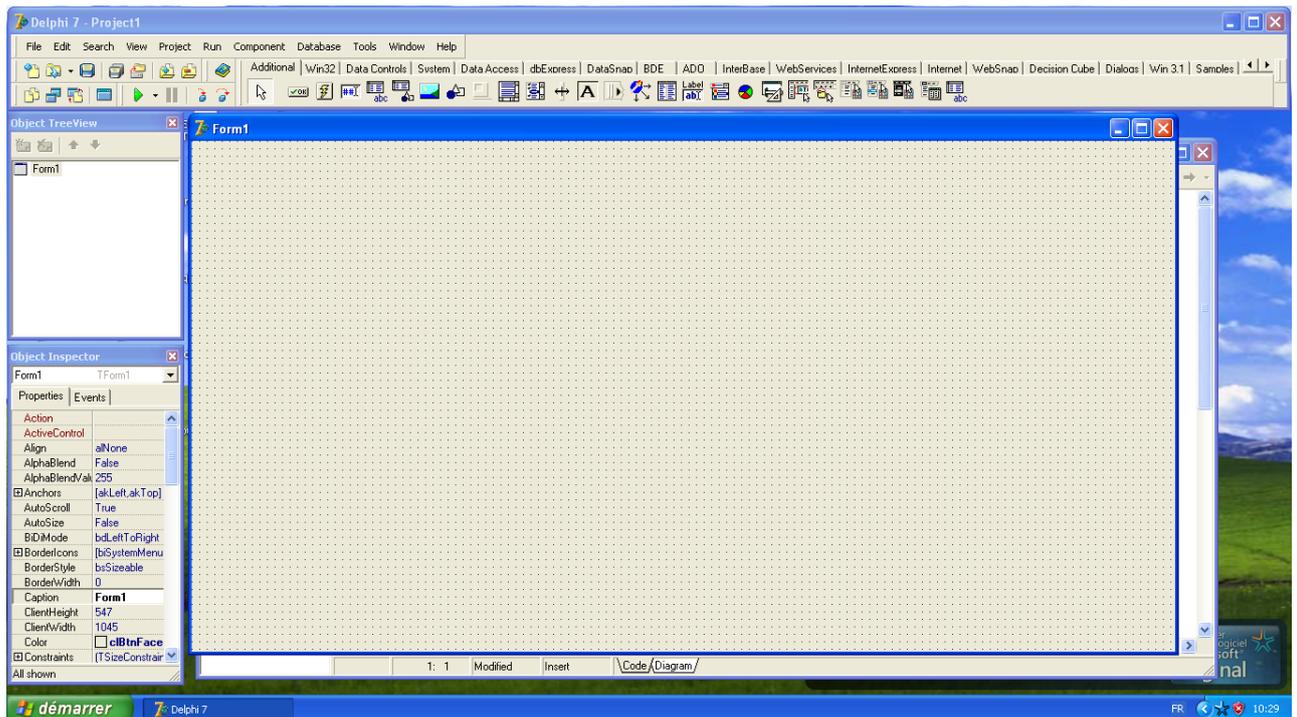


Figure III.2 :Delphi7

III. Description de l'application

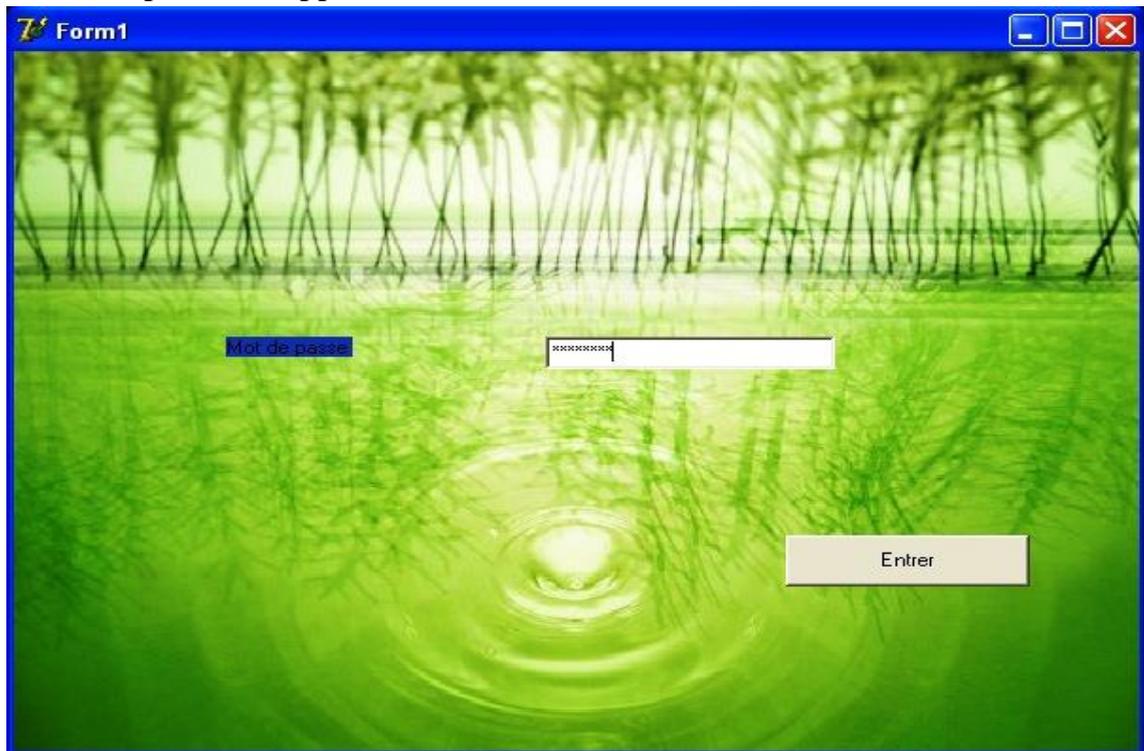


Figure III.3 : Authentification

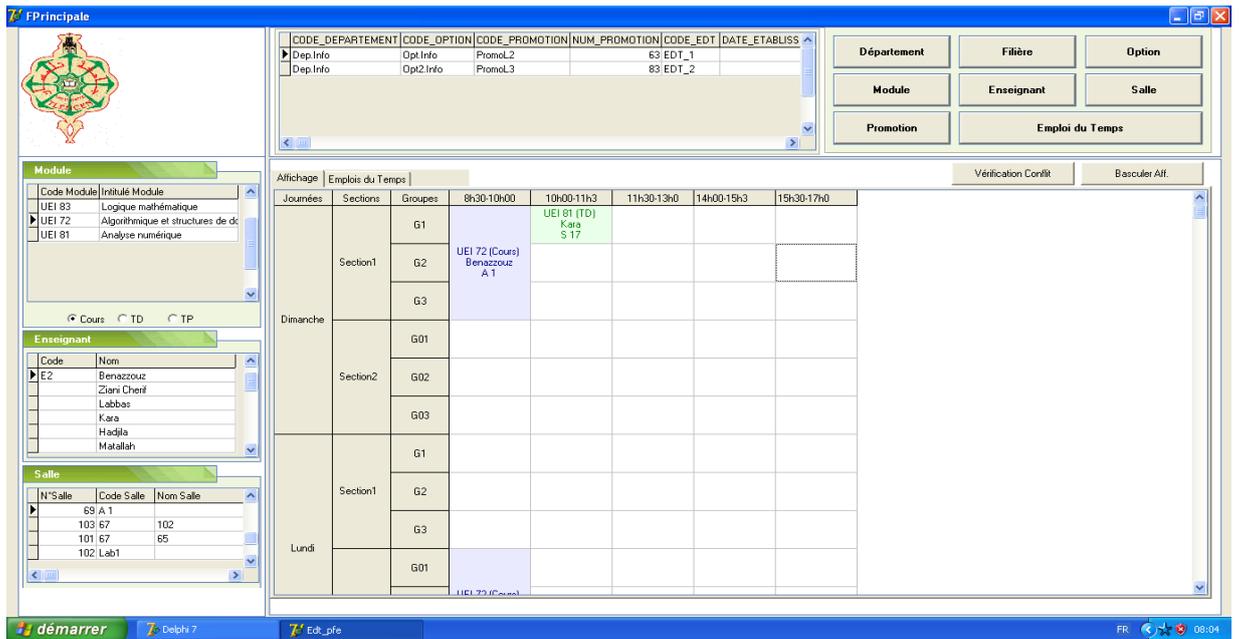


Figure III.4 : Interface de l'application

- Le tableau qui se trouve dans la partie supérieure présente les informations des différents emplois du temps (Code, Département, la promotion, Date d'établissement, Année Universitaire...).
- La partie qui englobe des Boutons « Département » « Promotion » « Filière » « Option » Chaque bouton conduit à une interface secondaire où cette interface représente des attributs de chaque table et aussi visualise les données de celle-ci.
- La figure suivante représente un exemple de cette interface :

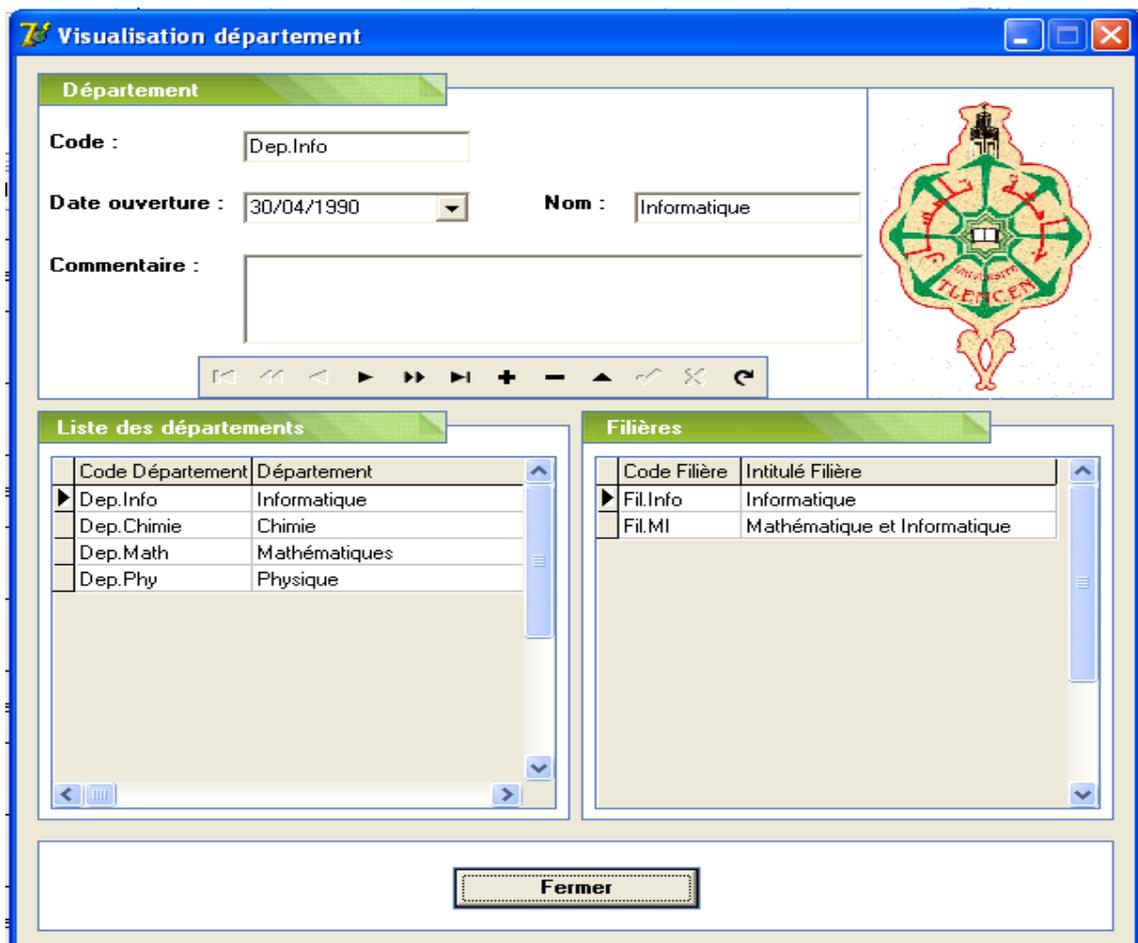


Figure III.5 : Modification table Département

Dans cette interface secondaire, on peut faire des opérations sur la table concernée comme Insérer, modifier ou supprimer une nouvelle donnée ou un nouvel attribut.

Dans une autre interface secondaire comme celle de « Visualisation Module », on peut aussi associer ou ajouter ou à chaque donnée d'autres données qui le comporte.

La figure suivante présente l'interface qui répond à ce besoin :

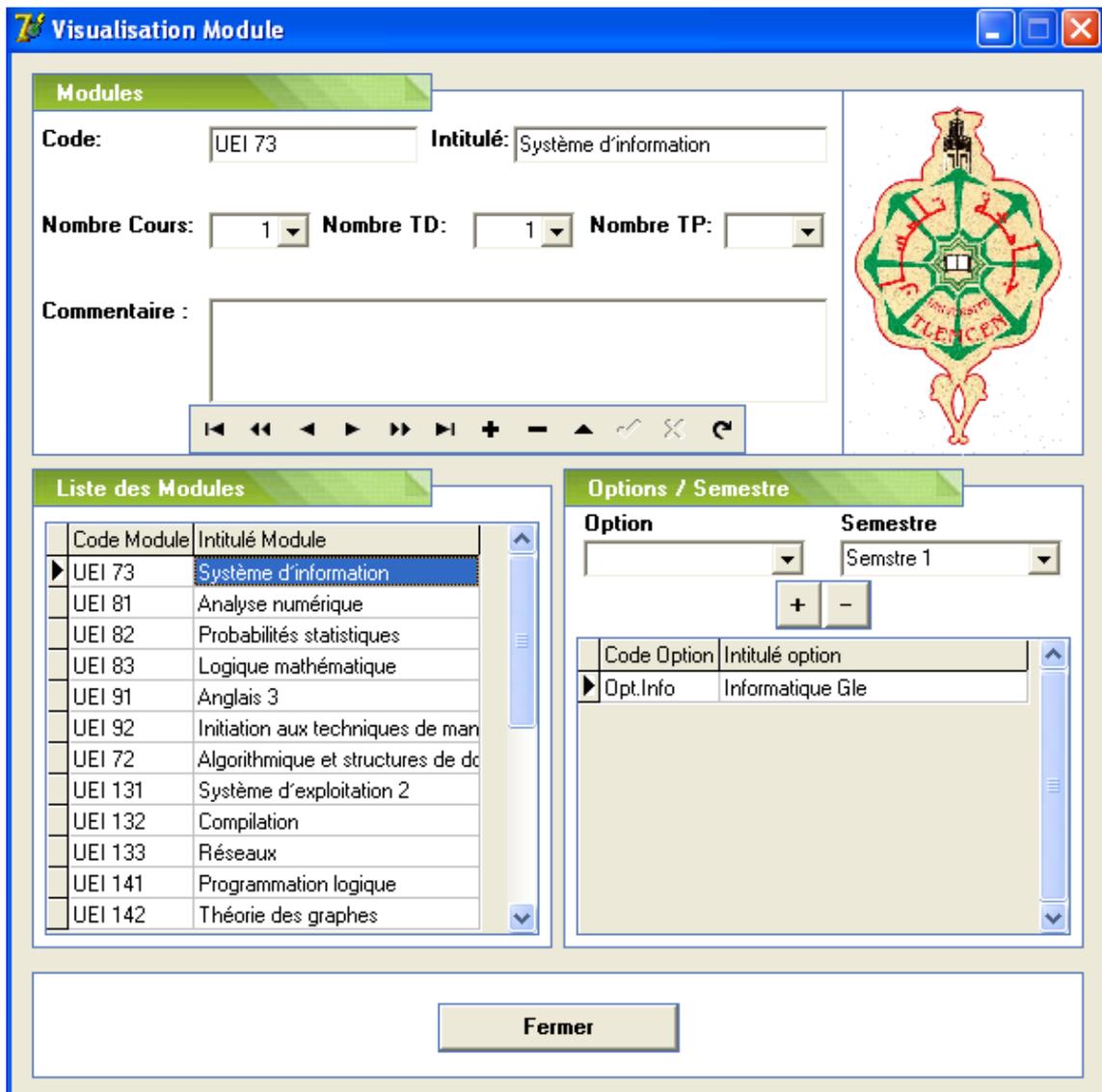


Figure III.6 : Modification table Module

- Dans la partie à gauche de l'interface
 - 1) Le tableau des modules comporte les données ou on les utilise pour remplir le tableau d'emploi du temps.
 - 2) Le Composant qui présente trois choix pour le type de la séance.
 - 3) Le tableau des Enseignants.
 - 4) Le tableau des salles Pour remplir une cellule donnée dans l'emploi du temps.

On double clic sur chaque tableau cité ci-dessus, ainsi, les informations de chaque ligne sont insérées.

La partie centrale de l'interface principale comporte deux onglets :

- 1) La première page « Affichage » est l'interface importante de notre application, elle sert à l'affichage d'un emploi du temps.

- 2) La deuxième page permet d'afficher l'emploi du temps de chaque enseignant à part (on affiche l'emploi du temps de l'enseignant concerné)

The screenshot shows a software window titled "FPrincipale" with a sidebar on the left containing filters for "Module", "Enseignant", and "Salle". The main area displays a table of EDT lines. The table has the following columns: NUM_GROUPE, NUM_SALLE, NUM_JOUR, NUM_HORAIRE, NUM_ENSEIGNANT, NUM_MODULE, NUM_LIGNE_EDT, and TYPE_SEANCE. The data rows are as follows:

NUM_GROUPE	NUM_SALLE	NUM_JOUR	NUM_HORAIRE	NUM_ENSEIGNANT	NUM_MODULE	NUM_LIGNE_EDT	TYPE_SEANCE
62	69	1	43	161	86	369	Cours
63	69	1	43	161	86	370	Cours
61	62	1	44	164	88	398	TD
66	69	2	43	161	86	470	Cours
65	69	2	43	161	86	469	Cours
61	69	1	43	161	86	472	Cours
64	69	2	43	161	86	468	Cours

Figure III.7 : Table des lignes d'EDT

En haut de ces deux pages, le Bouton « Basculer Aff. » sert à permuter disposition d'affichage de l'emploi du temps, En effet, et pour plus de lisibilité nous avons implémenté deux types d'affichage :

- 1- Un affichage dont les horaires sont en haut (en ligne).
- 2- Un affichage dont les horaires sont à gauche (en colonne).

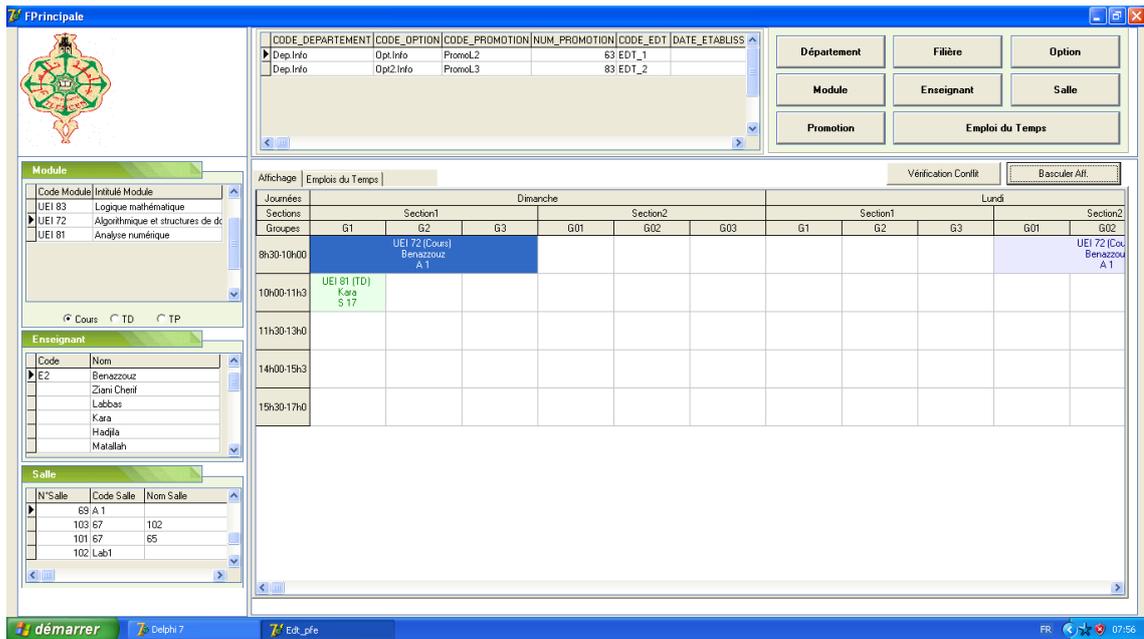


Figure III.8 : changement d’affichage d’EDT

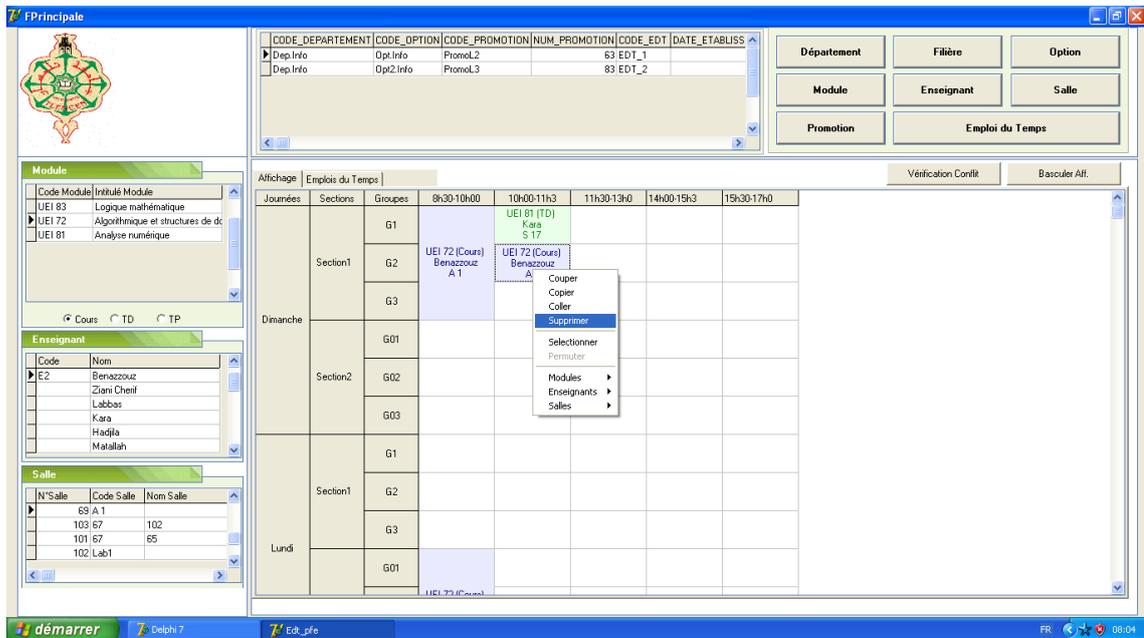


Figure III.9 : Modification d’EDT

- Le menu contextuel permet de faire des mises à jour d’une manière rapide et facile sur l’emploi du temps. Ce menu contient les options « copier, coller, couper, sélectionner, permuter.. » Il donne l’autorisation de copier, couper ou supprimer une cellule et permuter entre deux cellules.

Le bouton « Vérification Conflit » : sert à détecter un conflit de salle ou d’enseignant et affiche un message d’erreur s’il y en a de conflit.

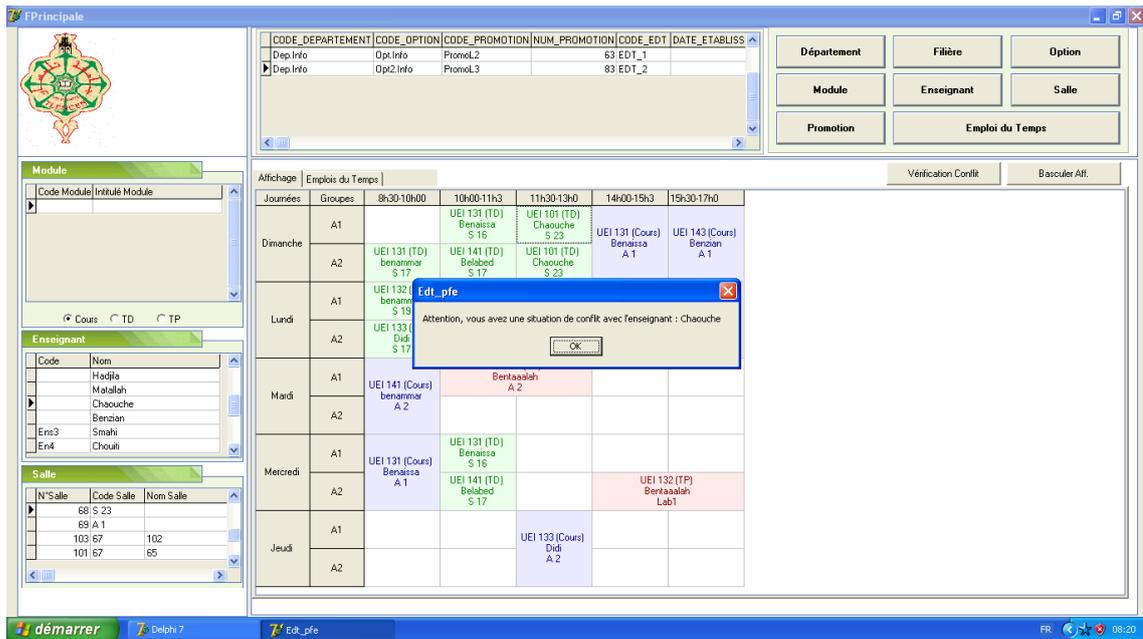


Figure III.10: Affichage d'un message d'erreur en cas de conflit

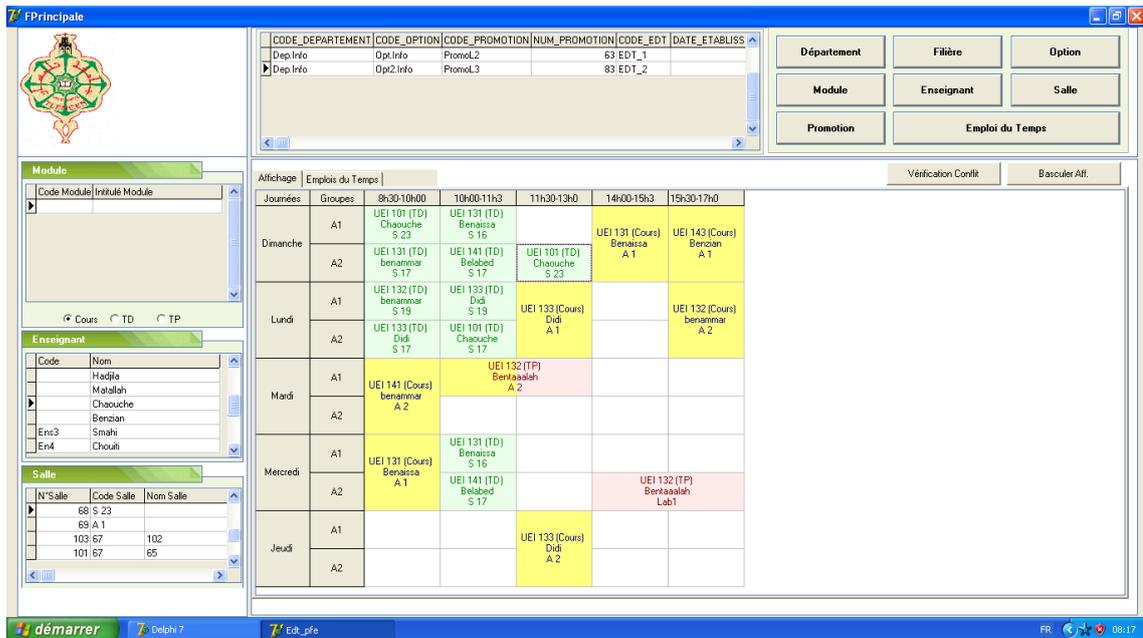


Figure III.11 : Affichage d'un EDT qui contient des conflits

IV. Conclusion :

Après le passage par les différentes étapes de développement, l'application a abouti à un logiciel fonctionnel qui répond globalement aux critères imposés pour notre projet.

Conclusion Générale

Durant les derniers trois mois, nous avons analysé et réalisé un projet de fin d'étude relatif à l'information de la génération d'emplois du temps au sein du département d'informatique De l'université Abou Bakr Belkaid Tlemcen (UABT).

Dans ce projet, nous avons exploité nos efforts et nos connaissances acquises durant notre formation pour pouvoir bien analyser le sujet de la génération d'emplois du temps. Nous espérons que ce travail peut ramener un plus au département.

Ainsi, nous avons appris les différentes étapes de conception d'un système d'information en utilisant le langage UML. D'un autre coté ce travail nous a permis d'acquérir des connaissances dans le domaine de la programmation et de conforter nos connaissances en conception logicielle. Ainsi que la programmation avec des bases des données SQL.

Perspectives :

Notre principal objectif était de faciliter l'établissement des emplois du temps au sein de notre département. Et comme tout projet nous souhaitons que nous aurons l'occasion de le développer et de l'enrichir pour plus d'efficacité, selon les perspectives suivantes :

- L'extension du system aux autres départements et facultés.
- Utiliser les méthodes d'optimisations pour résolutions les problèmes génération d'EDT (Emploi Du Temps)
- Injecter un mécanisme qui permet le traitement automatique et intelligent pour la génération « automatique » de l'emploi du temps en prenant en considération les préférences et les contrainte des enseignants.
- Améliorer l'interface du système.

Référencement bibliographie :

[1]: Jean-Patrick Matheron " Comprendre Merise"- Outils conceptuels et organisationnels ,5e édition (15 octobre 2002)

[2]: -Jean-Louis Le Moigne "Théorie De La Modélisation"- 01/06/1990 [Broché]

[3]: Marylène Micheloud et Medard Rieder, Programmation orientée objets en C++: *Une approche évolutive*, PPUR presses polytechniques, 2002

[4]: <https://www.delphi.developer.com>

[5]: Pascal Roques (Auteur), Franck Vallée (Auteur) - Etude (broché). Paru en 03/2007