

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعة عمار ثليجي بالاغواط

UNIVERSITE AMAR TELIDJI LAGHOUAT



كلية العلوم

FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

PROJET DE FIN D'ETUDE

(Licence)

Domaine : Mathématiques et Informatique

Filière : Informatique

Présenté par :

GUETTAF AZIZA

TRICHE ZAHIRA

THÈME

**Conception et réalisation d'un système de gestion des emplois
du temps**

Noms du membre de jury :

M.Maïcha Habib

Encadreur

M.Chellama Laaredj

Examinateur

Année Universitaire 2015/2016

Dédicaces

Je dédie ce travail

A ceux qui sont dans mon coeur, qui ont veillés pour notre confort et sacrifié beaucoup pour notre réussite, Ma chère mère (que dieu me la garde).

A celui qui m'a toujours appris comment réfléchir avant d'agir, à celui qui m'a soutenu tout au long de ma vie scolaire, à celui qui n'a jamais épargné un effort pour mon bien,

Mon cher père (Que dieu me le garde).

A mes chers frères , mes chères soeurs, toutes ma familles, mes chères amis .

A tous les membres de la promotion licence informatique de l'université **Ammar**

Telidji.

Et à tous ceux qui me connaissent .

ZAHIRA

Dédicaces

A ma mère

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi
.Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne
jamais vous décevoir.

Tu m'as donné la vie, le courage pour réussir .Tout ce que je peux t'offrir ne pourra
exprimer l'amour et la reconnaissance que je te porte.

Aucune dédicace ne serait exprimer assez profondément ce que je ressens envers vous .
Je vous dirais tout simplement , un grand merci ,je vous aime

A mon père

L'épaule solide, l'oeil attentif compréhensif et la personne la plus digne de mon estime et
de mon respect.

Aucune dédicace ne saurait exprimer mes sentiments, que Dieu te préserve et te procure
santé et longue vie.

A ma soeur et mon frère

Vous occupez une place particulière dans mon coeur je vous dédie ce travail en vos
souhaitant un avenir radieux plein de bon heur.

A mes très chers amis

En souvenir de nos éclats de rire et des bons moments .En Souvenir de tout ce qu'on à
vécu ensemble.

J'espère de tout mon coeur que notre amitié durera éternellement.

AZIZA

Remerciement

Nous remercions **ALLAH** le tout puissant, maître des cieux et de la terre, qui nous a éclairé le chemin et nous à permis de mener à bien ce travail.

Tout d'abord nous tenons surtout à adresser nos plus vifs remerciements à monsieur **Maïcha habib**, qui nous a accepter de travailler sous sa direction. Nous ne saurons jamais oublier ses conseils judicieux.

Un merci pour tous nos enseignants .

Un grand merci à toutes les personnes qui nous ont soutenues de près ou de loin au cours de la réalisation de ce modeste travail.

Nos remercie à tous les membres du jury qui ont accepté de juger notre travail.

Resumé

La génération automatique des emplois du temps universitaire peut être décrite comme l'allocation des ressources (Créneaux Horaires, Salles, ...) aux événements (Séances de cours, TD et TP), tout en essayant de satisfaire un ensemble de contraintes. Nous visons à travers ce travail à donner une application qui permette de résoudre les problèmes d'affectation de temps la plus complète possible en démarrant par une conception solide.

Mots Clés : Optimisation ,contrainte, Emploi du temps, Multi-Agents .

ملخص

تعتبر جدولة الأحداث داخل أي هيئة من أهم تطبيقات البرمجة في الإعلام الآلي، خاصة إذا كانت هذه الهيئة تتميز بكثرة الأحداث و تنوعها مثل الجامعة.

و يتمحور هذا البحث حول معالجة جدولة مواعيد الجامعة المختلفة ، و هذا باعتبار هذه العملية كعملية تخصيص للموارد (الوقت ، القاعة ، ...) للأحداث و هي (الدروس ، أعمال تطبيقية ، أعمال موجهة) مع التقييد بمجموعة من القيود ، مع البحث عن أحسن وسيلة لتوزيع الحصص على المجموعات و الأساتذة و القاعات

الكلمات المفتاحية

تحسين،التضاربات،الجدول الزمني،متعدد الوكلاء

Abstract

Automatic generation of university Timetabling can be described as the allocation of resources (time slots, rooms ...) to events (lesson, **TD** and **TP**), while trying to meet a set of constraints. The proposed approach is solving the problem using a special algorithm to create the Timetabling, With the search for the best way to distribute rations to groups , teachers and rooms without inconsistencies.

Keywords : Optimizing, compelled, Schedule, Multi Agents .

Table des matières

Introduction générale	1
1 L'emploi du temps	3
1.1 Introduction	4
1.2 Qu'est-ce qu'un emploi du temps ?	4
1.2.1 Définition de l'emploi du temps	4
1.2.2 Différentes variantes pour la formulation d'emploi du temps	5
1.2.2.1 Emploi du temps des écoles :	5
1.2.2.2 Emploi du temps des cours :	5
1.2.3 Les domaines d'application :	5
1.3 Génération automatique d'un emploi du temps	6
1.3.1 Définition formelle du problème de l'emploi du temps	6
1.3.2 Définition formelle	6
1.3.3 Formulation des contraintes	7
1.3.4 le PET comme problème de satisfaction des contraintes	9
1.4 Quelques méthodes utilisées pour résoudre le PET :	10
2 Modélisation	12
2.1 Introduction	13
2.2 Conception	13
2.3 représentation générale d'UML	13
2.3.1 définition	13
2.3.2 Présentation générale des diagrammes	14
2.3.2.1 Vues statiques	14
2.3.2.2 Vues dynamiques	15

2.3.3	les objectifs d'UML :	15
2.4	Diagramme de cas d'utilisation global :	16
2.5	Diagramme de cas d'utilisation :Authentification	16
2.6	Diagramme de séquence Authentification	17
2.7	Diagramme de cas d'utilisation :Saisie	18
2.8	Diagramme de séquence Saisie	18
2.9	Diagramme de cas d'utilisation :Enregistrer	19
2.10	Diagramme de séquence Enregistrer	20
2.11	Diagramme de cas d'utilisation :Enregistrer sous	21
2.12	Diagramme de séquence Enregistrer sous	21
2.13	Diagramme de cas d'utilisation : Imprimer	22
2.14	Diagramme de séquence Imprimer	23
2.15	Diagramme de classe	24
2.16	Modèle relationnel(MLD)	24
2.17	Dictionnaire des données	25
2.17.1	définition	25
3	Réalisation et aspect d'implémentation	28
3.1	Introduction	29
3.2	l'enviennement de programmation JAVA	29
3.3	l'environnement de gestion de BDD MYSQL	29
3.4	L'environnement de traitement de texte LATEX	30
3.4.1	Définition de LATEX	30
3.4.2	Quelques caractéristiques :	30
3.5	Présentation l'application	30
	Bibliographie-Webographie	36

Table des figures

2.1	<i>Diagramme de cas d'utilisation :global</i>	16
2.2	<i>Diagramme de cas d'utilisation :Authentication</i>	16
2.3	<i>Diagramme de séquence Authentication</i>	17
2.4	<i>Diagramme de cas d'utilisation :Saisie</i>	18
2.5	<i>Diagramme de séquence Saisie</i>	18
2.6	<i>Diagramme de cas d'utilisation Enregistrer</i>	19
2.7	<i>Diagramme de séquence Enregistrer</i>	20
2.8	<i>Diagramme de cas d'utilisation :Enregistrer sous</i>	21
2.9	<i>Diagramme de séquence Enregistrer sous</i>	21
2.10	<i>Diagramme de cas d'utilisation : Imprimer</i>	22
2.11	<i>Diagramme de séquence Imprimer</i>	23
2.12	<i>Diagramme de classe</i>	24
3.1	<i>Interface de connection au système</i>	30
3.2	<i>Interface principale</i>	31
3.3	<i>Interface principale après la saisi</i>	32
3.4	<i>Interface prof_module</i>	33
3.5	<i>Interface prof_module après insertion</i>	33
3.6	<i>Interface prof_module après la retirer</i>	34
3.7	<i>Interface de sauvegarde</i>	34

Liste des tableaux

2.1	<i>Descriptions des cas d'utilisation Authentification</i>	17
2.2	<i>Descriptions des cas d'utilisation Saisie</i>	19
2.3	<i>Descriptions des cas d'utilisation Enregistrer.</i>	20
2.4	<i>Descriptions de cas d'utilisation Enregistrer sous.</i>	22
2.5	<i>Descriptions de cas d'utilisation Imprimer.</i>	23
2.6	Dictionnaire de donnée	27

Introduction générale

Une chose que nous ne pourrons jamais la rattraper après qu'elle soit passé, c'est le temps .En conséquence, ce qui est fait est fait, alors la gestion du temps est une opération très délicate est devient parfois immaîtrisable.

En se basant sur le principe du célèbre proverbe qui dit ” **mieux vaut prévenir que guérir** ”, on va proposer une solution pour éviter au maximum de se trouver dans des erreurs liées à la gestion du temps.

Cependant, l'homme à montrer sa faiblesse face à des fastidieux travaux à la fois complexes et difficiles, et cela du à l'insuffisance de ses capacités physiques et intellectuelles. L'être humain a crée des machines manuelles par la suite des machines automatiques, des outils et beaucoup d'autres choses pour minimiser son intervention dans le processus de travail. Pour but de chercher des moyens pour faciliter la vie de l'être humain. Notre but est d'automatiser les tâches pour : aider l'homme dans les prises de décision, traiter les informations avec précision... Les solutions trouvées à ce problème sont nombreuses et acceptable, mais notre but est de chercher une amélioration à ces solutions pour minimiser l'intervention de l'Homme dans le processus de travail.

Dans notre mémoire on présente une étude d'un problème classé parmi les problèmes les plus complexes (**NP-complet**) dont la solution est très difficile et nécessite des approches un peu particulières.[1]

Notre travail consiste à mettre au point un système de génération automatique d'un emploi du temps. Ce problème se pose dans tous les secteurs professionnels et sa solution apporte une bonne gestion du temps.

L'emploi de temps est l'affectation des personnes et matérielles dans un certain nombre de créneaux horaires en respectant les contraintes. Cette affectation a pour but de bien organiser le travail et d'éviter les conflits et de construire un environnement de travail

adéquat.

Cependant la réalisation d'un emploi de temps manuellement est très compliquée et très coûteuse. Beaucoup de chercheurs sont intéressés par ce type de problèmes et proposent leurs solutions mais chacun d'eux a ses propres méthodes.

Notre document est organisé comme suit :

Le premier chapitre présente une description synthétisée au problème d'emploi du Temps on a commencé par l'identification et la description du problème d'emploi du temps, de la génération automatique d'emploi du temps. Outre on a cité quelques domaines d'application, les différentes variantes pour la formulation d'emploi du temps, les contraintes d'emploi du temps et environnement, puis on a présenté quelques méthodes utilisées pour résoudre le problème d'emploi du temps .

Le deuxième chapitre contient l'étude analytique qu'on a fait pour la résolution du problème, et on a montré toutes les schéma UML, du remplissage des informations de la base de données correspondant au faculté ainsi les autres facteurs nécessaires (départements, domaines, matières, groupes, enseignants, emplacement).

Le troisième chapitre est consacré à l'implémentation de notre application .

Chapitre 1

L'emploi du temps

1.1 Introduction

L'emploi du temps est l'un des exemples les plus fréquents des problèmes d'optimisation dans une institution ou un établissement. IL peut se manifester sous plusieurs formes différentes et particulières selon la spécificité de l'environnement ou l'institution où l'on en a besoin.

Réaliser un emploi du temps pose un problème qu'on doit résoudre. D'une manière générale pour résoudre un problème donné on commence par la recherche de la classe à laquelle appartient ce problème. Le problème d'emploi fait partie des problèmes dont les contraintes augmentent en fonction de la résolution, ce qui rend difficile leurs résolutions. De tels problèmes peuvent être résolus par des approches classiques ou modernes.

Dans ce chapitre, nous venons de voir qu'est-ce qu'un emploi du temps (**EDT**), sa définition, ses domaines d'applications. Nous avons fait une description de l'**EDT** d'un établissement scolaire. Nous avons également : détaillé les divers problèmes liés à l'**EDT**, indiqué comment formuler le **PET**, défini une génération automatique d'un **EDT** ses avantages par rapport à la réalisation manuelle d'un **EDT**.

Nous avons conclu que la résolution d'un problème d'emploi du temps est une gestion des contraintes.

1.2 Qu'est-ce qu'un emploi du temps ?

1.2.1 Définition de l'emploi du temps

L'emploi du temps est un plan représentatif définissant le temps nécessaire pour élaborer des tâches et des objectifs fixés préalablement sous forme de créneaux horaires. Il consiste à gérer les charges de travail dans le temps tout en prenant compte des ressources humaines et matérielles disponibles.

Donc on peut définir le problème lié à la gestion de l'emploi du temps comme chercher à gérer les ressources d'une manière satisfaisante en respectant au maximum que possible les contraintes de temps.

1.2.2 Différentes variantes pour la formulation d'emploi du temps

Il existe un grand nombre de variantes du problème d'emploi du temps qui diffère les uns des autres selon le type d'établissement impliqué (université ou école...) et le type de contraintes imposées.

Parmi plusieurs classifications possibles de l'emploi du temps nous avons choisi une classification en deux catégories principales :

1.2.2.1 Emploi du temps des écoles :

Établir un programme journalier pour toutes les classes d'une école en évitant que deux professeurs se rencontrent en même temps dans une même salle et que deux salles soient attribuées à un professeur en même temps.

1.2.2.2 Emploi du temps des cours :

Établir le programme hebdomadaire pour tous les cours d'un ensemble d'université en réduisant au minimum les chevauchements des cours ayant les étudiants communs.

Cependant, une telle classification n'est pas stricte dans le sens qu'il y a quelques problèmes spécifiques et ne pouvant être facilement classés dans les deux catégories citées ci-dessus. Par exemple l'emploi du temps d'une école spécifique dans laquelle elle donne la grande liberté à l'étudiant, l'ensemble de cours peut être semblable à un emploi du temps de cours.

En pratique certaines conditions s'imposent pour l'élaboration d'un emploi du temps. Le système doit spécifier l'affectation des enseignants, groupes, salles et matérielles pédagogiques dans le temps tout en respectant le plus possible les contraintes imposées.

1.2.3 Les domaines d'application :

Le problème de l'emploi du temps se pose en général dans tous les secteurs : économiques, industriels, administratifs, les établissements scolaires, les instituts, les universités, les entreprises...etc.

Les résultats issus d'un secteur peuvent être utilisés dans un autre secteur mais nous nous intéressons particulièrement qu'aux établissements scolaires.

1.3 Génération automatique d'un emploi du temps

La génération automatique d'un emploi du temps est une activité de création, de gestion et de maintenance d'un emploi du temps avec un ordinateur et avec l'intervention minimale de l'être humain en satisfaisant au mieux les ressources humaines, matérielles et temporelles.

La génération automatique d'un emploi du temps nécessite la construction d'un système capable de gérer les ressources temporelles et matérielles, les contraintes imposées et de résoudre les conflits entre ces ressources.

Mettre au point un tel système est une tâche difficile et fastidieuse pour l'être humain. Outre la réalisation manuelle d'un emploi de temps est très compliquée, très coûteuse en terme de temps et effort mentale.

La génération d'un emploi du temps n'est pas un problème de simulation car il ne s'agit pas de recréer virtuellement le comportement d'un <organisme> existant mais de fournir plutôt une résolution de problèmes réels.

1.3.1 Définition formelle du problème de l'emploi du temps

Un problème de l'emploi du temps (**PET**) est défini par la gestion d'un ensemble de ressources (matérielles, humaines et temporelles) et la satisfaction d'un ensemble des contraintes.

Le **PET** consiste à concilier des contraintes temporelles et les ressources pour proposer un emploi du temps sur une certaine durée entre plusieurs acteurs (professeurs, étudiants, cours, salles ...).

Dans ce cas les buts sont préétablis en fonction d'un plan pédagogique et des contraintes portant sur :

- Des disponibilités temporelles ;
- Des compétences ;
- Le besoin des ressources spécifiques (matériels pédagogiques).

1.3.2 Définition formelle

Formellement on peut représenter le problème de l'emploi du temps comme suit :

- un ensemble de classes : c_1, \dots, c_m ; avec m est le nombre de salles,

- un ensemble de professeurs : t_1, \dots, t_n ; avec n est le nombre de professeurs ;
- un ensemble de périodes de $1, \dots, h$ avec h est le nombre de périodes.

Une triplet (X_{ijk}) indique les contraintes entre les éléments (i : salle, j : professeur, k : cours).

La Formulation mathématique :

Résoudre **PET** : consiste à trouver :

$$X_{ijk} \begin{cases} 1 & \text{si la classe } C_i \text{ et } \leq \text{ professeur et } j \text{ seréunt à la période } hk \\ 0 & \text{dans le cas contraire.} \end{cases}$$

Une triplet (X_{ijk}) indique les contraintes entre les éléments (i : salle, j : professeur, k : cours).

La Formulation mathématique :

Résoudre **PET** : consiste à trouver

1.3.3 Formulation des contraintes

Un horaire peut être considéré comme satisfaisant s'il convient aux professeurs, aux élèves et s'il respecte la disponibilité des salles. Le souci d'assurer un certain confort aux professeurs et aux élèves rend particulièrement difficile la confection d'un emploi du temps.

Il est bien difficile de recenser toutes les exigences susceptibles. D'intervenir dans un problème d'emplois du temps.

Certaines découlent directement des données décrites dans la section précédente alors que d'autres viennent se superposer selon le cas.

En clair, le problème de la confection d'un emploi du temps universitaire consiste à donner une heure de début j **P** à chaque leçon **up i up 1, ..., up n** tout en satisfaisant les contraintes suivantes :

1. **Non chevauchement des Intervenants** : un intervenant ne peut pas enseigner plus d'une leçon à la fois.
2. **Contraintes de site pour Intervenants** : L'enseignant ne change pas de site au cours d'une demi-journée.
3. **Contraintes de pause pour Intervenants** : L'enseignant a au moins une pause de 45 minutes aux heures de midi.

4. **Disponibilité des Intervenants** :L'emploi du temps est conçu d'une façon en essayant de respecter la journée de disponibilité des enseignants. Aucune différence n'est faite entre les intervenants extérieurs et le personnel de l'université.
5. **Non chevauchements des étudiants** :Un étudiant ne peut suivre qu'une seule unité pédagogique à la fois.
6. **Contraintes de pause Étudiants** :L'étudiant a au moins une pause de 45 minutes aux heures de midi.
7. **Contraintes sur les horaires des UPs** :On ne peut commencer une unité pédagogique si on ne peut la finir dans les horaires prédéfinis. Par exemple, on ne peut commencer un cours de deux heures à 19h.
8. **Contraintes de positionnement des UPs** :Toutes les unités pédagogiques doivent être positionnées.
9. **Contraintes d'affectation de salles** : Dans une salle, il ne peut y avoir qu'une seule unité pédagogique programmée par créneau horaire.
10. **Contraintes de choix de salles** : les affectations des salles doivent respecter les demandes des responsables Valeur telque chaque **UV** est caractérisé par les informations(Un ensemble d'intervenants, des groupes, des filières,...).
11. **Contraintes de capacité de salle** :Une unité pédagogique ne peut être planifiée dans une salle ne permettant pas de recevoir l'ensemble des étudiants lié à cette unité.
12. **Contraintes jours non ouvrable** : aucun enseignement ne peut avoir lieu durant une journée non ouvrable.

L'existence d'un horaire respectant à la lettre toutes les contraintes n'est pas garantie dans le cas général. En réalité, il s'agit de trouver un horaire aussi bon que possible dans un temps de calcul raisonnable tout en tolérant la présence de conflits ; le nombre de ces conflits doit être minimisé pour que l'horaire soit jugé satisfaisant par les différents parties concernées (direction de l'université, Intervenants, Étudiants, etc.)

L'emploi du temps est un problème difficile à élaborer et plus demandé à tous les niveaux dans divers domaines. La difficulté principale vient de la complexité de gérer les contraintes temporelles qui le caractérisent. Donc La génération d'un emploi du temps et à la fois :

- Difficile car c'est typiquement un problème de résolution de contraintes dont la solution n'est pas à priori connue dans le cas général ;
- Et nécessite de fournir un programme capable de s'adapter avec les changements dynamiques de l'environnement pour fournir une solution meilleure.

1.3.4 le PET comme problème de satisfaction des contraintes

Les acteurs mis en jeu pour l'élaboration d'un emploi du temps au niveau scolaire sont :

- des enseignants ;
- des groupes d'étudiants ;
- des salles ;
- des matières ;

On peut classer les diverses contraintes en deux catégories : les contraintes fortes et les contraintes faibles. Qui doivent être satisfaites peuvent être violées. Chacun des acteurs possède des contraintes fortes et faibles.

a) Les contraintes des enseignants :

- **fortes :**
 - sa disponibilité (jour de la semaine, tranche horaire, ...);
 - ses propres qualifications ;
 - ses préférences ;
 - sa possibilité d'enseigner un ou plusieurs modules ;
- **faibles :**
 - ses compétences ;
 - son type de contrat (permanent ou contractuel) ;

b) Les contraintes des ressources matérielles :

- **fortes :**
 - disponibilité des salles et appareils (data show...);
- **faibles :**
 - équipements spéciaux (chauffage, ventilateur, climatiseur ...);

c) Les contraintes des enseignants :

- **fortes :**

— chaque module doit être enseigné au minimum par un enseignant ;

Le problème à résoudre consiste à concilier toutes ces contraintes pour proposer un emploi du temps sur une certaine durée.

Malgré les difficultés d'implémentation, il y existe plusieurs approches proposées et étudiées par de nombreux chercheurs, ce qui offre des solutions réalisables mais qui ne sont pas toujours efficaces.

1.4 Quelques méthodes utilisées pour résoudre le PET :

L'heuristique : L'heuristique est une règle générale d'action applicable à plusieurs situations permettant d'aboutir plus rapidement à la solution. Pour le cas de l'emploi du temps l'heuristique remplit directement l'horaire complet d'un cours ou d'un groupe de cours à la fois dans la mesure où aucun conflit ne surgit. Un système Schola est un exemple typique de cette méthode. Ce système est basé sur les trois stratégies suivantes :

- A** Assigner le cours le plus urgent à la période la plus favorable ;
- B** Quand une période peut être affectée seulement pour un cours, assigner une période à un cours ;
- C** Déplacer un cours déjà programmé à une période libre afin de laisser la période au cours cherchant à être affecté. Le système **SCHOLA programme** les cours en alternant les stratégies **A** et **B** autant que possible. Quand plusieurs cours ne peuvent être programmés de cette façon, elle commence à employer la stratégie **C**.

La stratégie **A** est le noyau du système et elle est utilisée dans plusieurs systèmes en définissant l'urgence et la période favorable différemment.

Un cours est urgent quand le professeur et la classe ont peu de disponibilité pendant qu'il y a beaucoup de cours à dispenser. Une période est favorable quand un petit nombre de cours peut être programmé à la période selon la disponibilité des professeurs et salles.

L'utilisation de la stratégie **B** est effectuée dans le cas où il y a un défaut au niveau de la stratégie **A**. La stratégie **C** fournit une forme limitée de retour arrière à récupérer les défauts de la stratégie **A**.

Conclusion

Dans cet chapitre, nous venons de voir ce qu'est un emploi du temps (**EDT**), sa définition, ses domaines d'applications. Nous avons fait une description de l'**EDT** d'un établissement scolaire.

Nous avons également : détaillé les divers problèmes liés à l'**EDT**, indiqué comment formuler le **PET**, défini une génération automatique d'un **EDT** ses avantages par rapport à la réalisation manuelle d'un **EDT**.

Nous avons conclu que la résolution d'un problème d'emploi du temps est une gestion de contraintes.

Chapitre 2

Modélisation

2.1 Introduction

Parmi les étapes les plus importantes de la programmation l'étape de conception, qu'on ne peut pas y dépasser et sans passer par cette étape on trouve des grandes erreurs dans les programmes.

Pour la conception des programmes plusieurs outils peuvent être utilisés, parmi ces outils le langage **UML** qui est choisi pour la conception de notre programme.

Parmi les multiples outils logiciels utilisés pour dessiner nos diagrammes, nous avons choisi

ArgoUML par ce qu'il est libre et gratuit. et en utilisant le dictionnaire de donnée pour la collection des données.

2.2 Conception

Dans ce qui suit, on va présenter la conception (les différents diagrammes) de notre projet en utilisant **UML**. On commence par la présentation de diagramme de cas d'utilisation générale, puis on détail chaque cas en donnant sa définition et son diagramme de séquence et enfin, on terminera par la présentation de diagrammes de classes du projet.

2.3 représentation générale d'UML

2.3.1 définition

Regardons tout d'abord ce qui s'est passé au début des années 90. Par rapport à la cinquantaine de méthodes d'analyse et de conception objet qui existaient au début des années 90, seulement trois d'entre elles se sont détachées nettement au bout de quelques années. En effet, la volonté de converger vers une méthode unifiée était déjà bien réelle et c'est pour cette raison que les méthodes **OMT**, **BOOCH** et **OOSE** se sont démarquées des autres.

OMT (Object Modeling Technique) de **James Rumbaugh** et **BOOCH** de **Grady Booch** ont été les deux méthodes les plus diffusées en France durant les années 90. Par ailleurs, **OOSE** de **Ivar Jacobson** s'est aussi imposée dans le monde objet pour la partie formalisation des besoins.

Pour aller plus loin dans le rapprochement, **James Rumbaugh** et **Grady Booch** se sont retrouvés au sein de la société Rational Software et ont été ensuite rejoints par **Ivar Jacobson** en se donnant comme objectif de fusionner leur méthode et créer **UML (Unified Methode Language)**.

Il est important de noter que contrairement à ce qui avait été envisagé au départ, le processus de développement a été sorti du champ couvert par le projet de norme. **UML** est donc une norme du langage de modélisation objet qui a été publiée, dans sa première version, en novembre 1997 par l'**OMG (Object Management Group)**, instance de normalisation internationale du domaine de l'objet.

En quelques années, **UML** s'est imposée comme standard à utiliser en tant que langage de modélisation objet.

Aujourd'hui, en cette fin de la première décennie des années 2000, nous avons déjà une dizaine d'années de recul sur l'enseignement et la pratique d'**UML** en entreprise.[2]

2.3.2 Présentation générale des diagrammes

UML propose **neuf** diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles :

2.3.2.1 Vues statiques

1. les diagrammes de classes

Permettent de spécifier la structure et les liens entre les objets dont le système est composé.

Réalise les fonctionnalités décrites par les diagrammes de cas d'utilisation.

2. les diagrammes de cas d'utilisation :

Montrer les limites d'un système et ses fonctions principales (pour les utilisateurs) à quoi sert le système en organisant les interactions possibles avec les acteurs.

3. Les diagrammes de composants :

Sont les diagrammes qui montrent les choix de réalisation, est utilisé que pour les systèmes complexes.

Un composant est un élément physique qui représente une partie implémentée d'un système qui peut être du code (source, binaire ou exécutable), un fichier de commande,

un fichier de données, une table

2.3.2.2 Vues dynamiques

1. Les diagrammes de séquence :

Permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs.

2. Les diagrammes de collaboration :

Il définit les communications qui existent entre objets au niveau global.

Les graphes de collaboration indiquent qui communique avec qui. Ils sont utilisés pour analyser et concevoir les structures de communications entre objets.

3. Les diagrammes d'états transition.

4. Les diagrammes d'activités.

2.3.3 les objectifs d'UML :

Objectifs poursuivis : [3]

- représenter des systèmes entiers (au delà seul logiciel) par des concepts objets ;
- créer un langage de modélisation utilisable par les humains et les machines ;
- établir un couplage explicite entre les concepts et les artefacts exécutables ;

2.4 Diagramme de cas d'utilisation global :

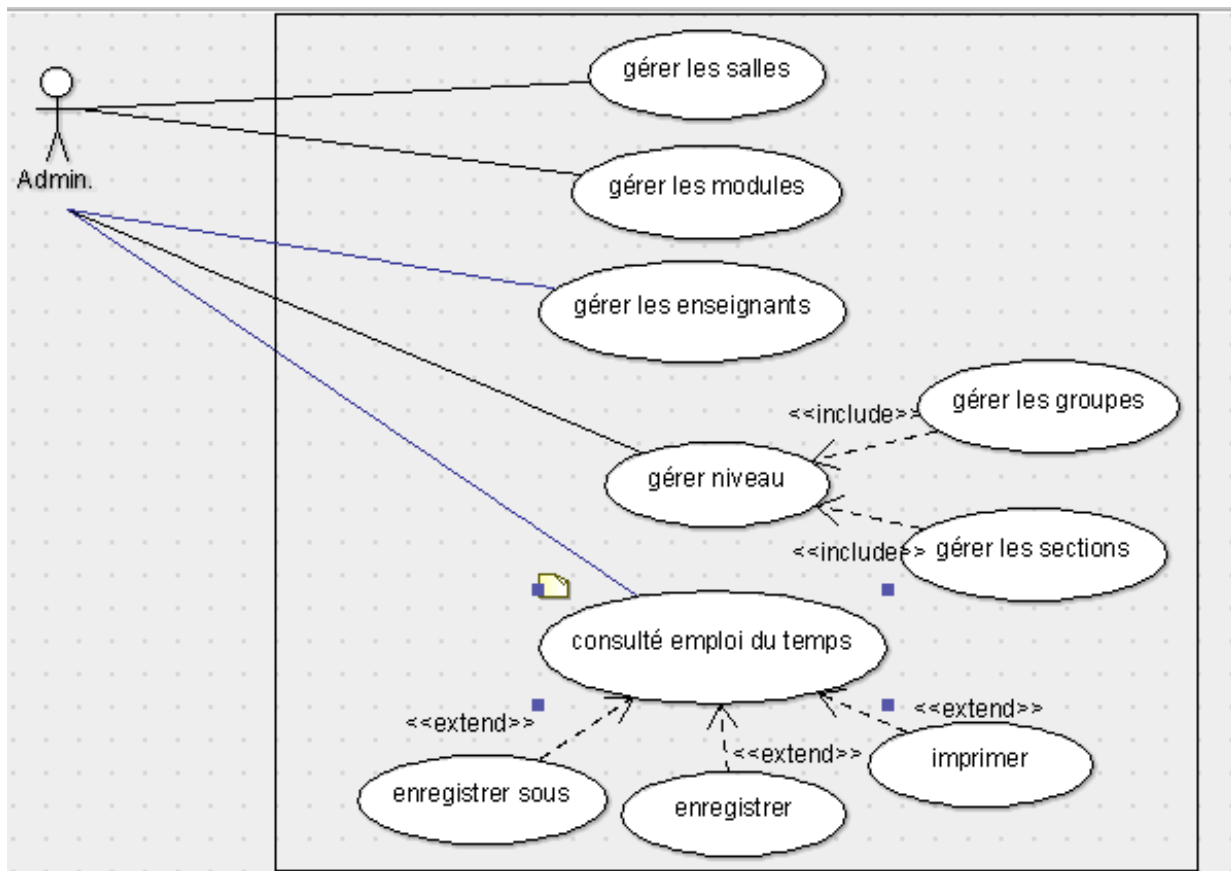


FIGURE 2.1: Diagramme de cas d'utilisation :global

2.5 Diagramme de cas d'utilisation :Authentication

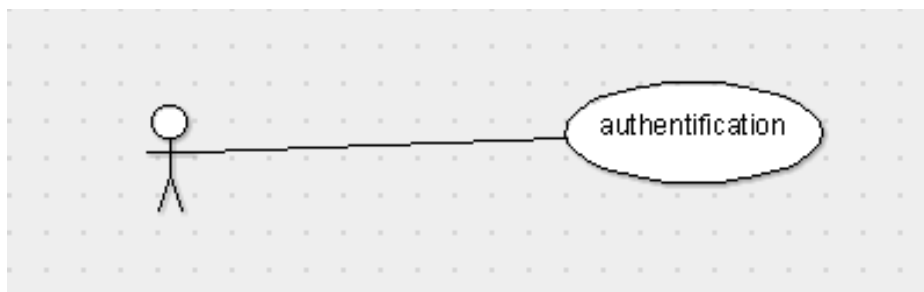


FIGURE 2.2: Diagramme de cas d'utilisation :Authentication

2.6 Diagramme de séquence Authentification

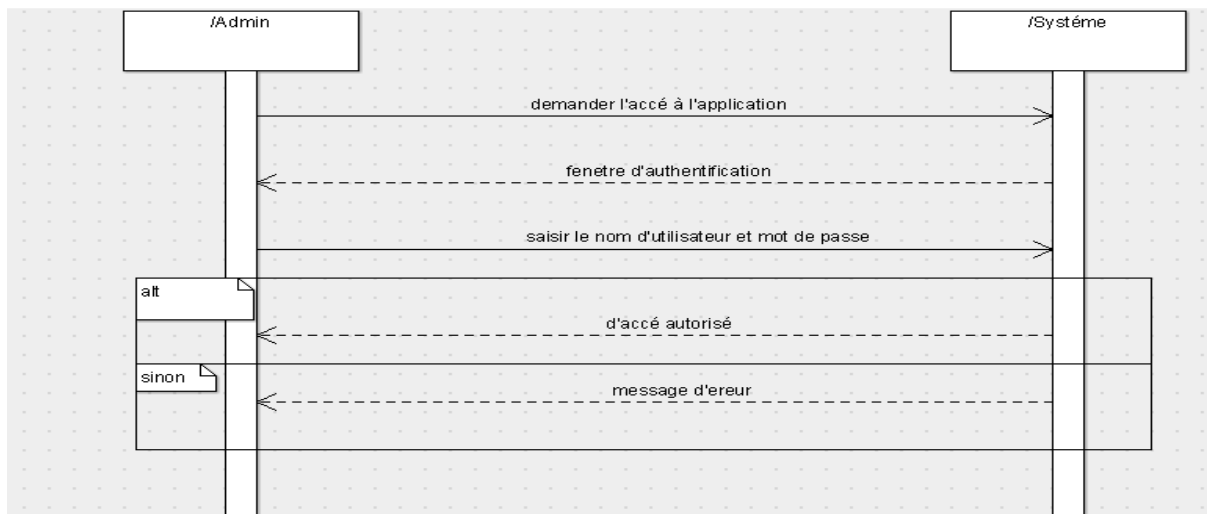


FIGURE 2.3: *Diagramme de séquence Authentification*

Description du sommaire	
Titre	Authentification
But	Ce cas d'utilisation permet à un administrateur de se connecter au Système.
Acteurs	administrateur.
Description du sommaire	
Enchainements :	<ol style="list-style-type: none"> 1. L'administrateur demande une connexion au système ; 2. Le système demande le nom d'utilisateur et le mot de passe ; 3. L'administrateur entre le nom d'utilisateur et le mot de passe ; 4. Le système vérifie l'existence de l'administrateur ; 5. Le système se connecte et ouvre la session ; <p>Exception 1 : dans le cas où l'administrateur saisie un mot de passe erroné, Le système demande de retaper les informations.</p>

TABLE 2.1: *Descriptions des cas d'utilisation Authentification*

2.7 Diagramme de cas d'utilisation :Saisie

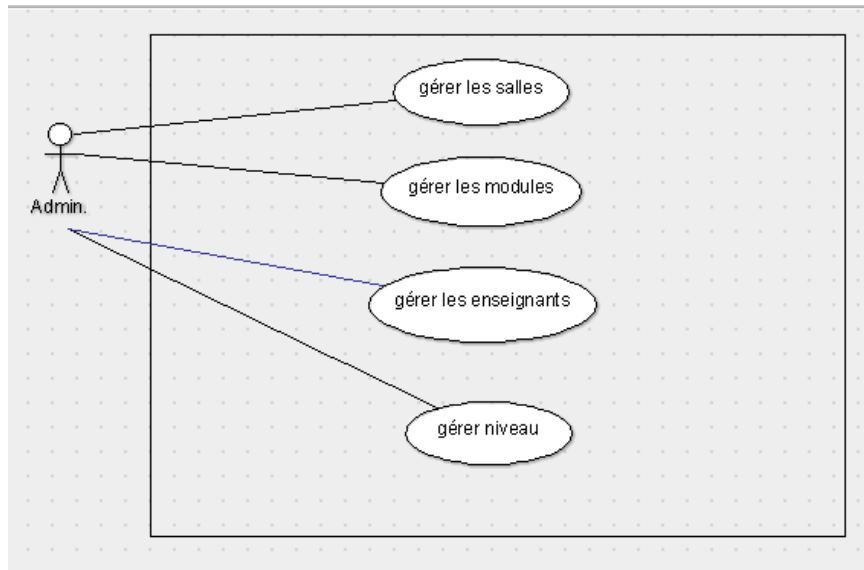


FIGURE 2.4: Diagramme de cas d'utilisation :Saisie

2.8 Diagramme de séquence Saisie

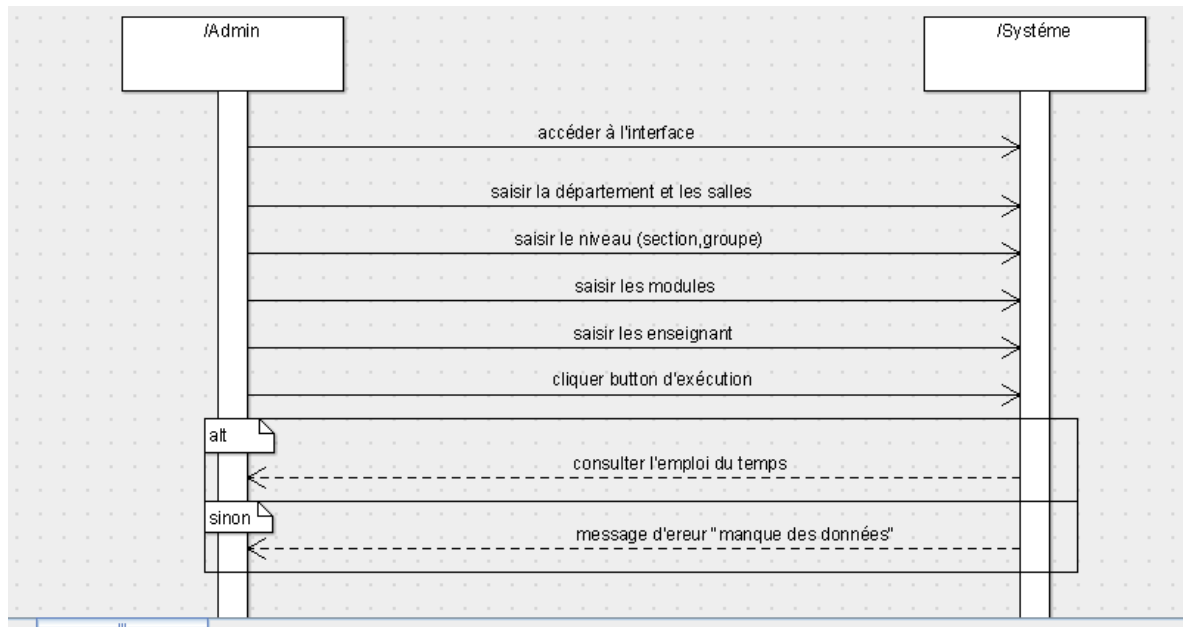


FIGURE 2.5: Diagramme de séquence Saisie

Description du sommaire	
Titre :	Saisie.
But :	Ce cas d'utilisation permet à un administrateur de saisir les informations suivants :(le département, les salles, le niveau (section, groupe), les modules, les enseignants).
Acteurs :	administrateur.
Description du sommaire	
Enchainements :	<ol style="list-style-type: none"> 1. l'administrateur accède à l'interface ; 2. l'administrateur saisie les informations suivantes : (le département, les salles, le niveau(section, groupe),les modules, les enseignants) ; 3. après la saisie de ces données l'administrateur enregistre le travail ; 4. le système demande de consulter l'emploi du temps ; <p>Exception : le système affiche un message d'erreur ?manque de données ?.</p>

TABLE 2.2: Descriptions des cas d'utilisation Saisie

2.9 Diagramme de cas d'utilisation :Enregistrer

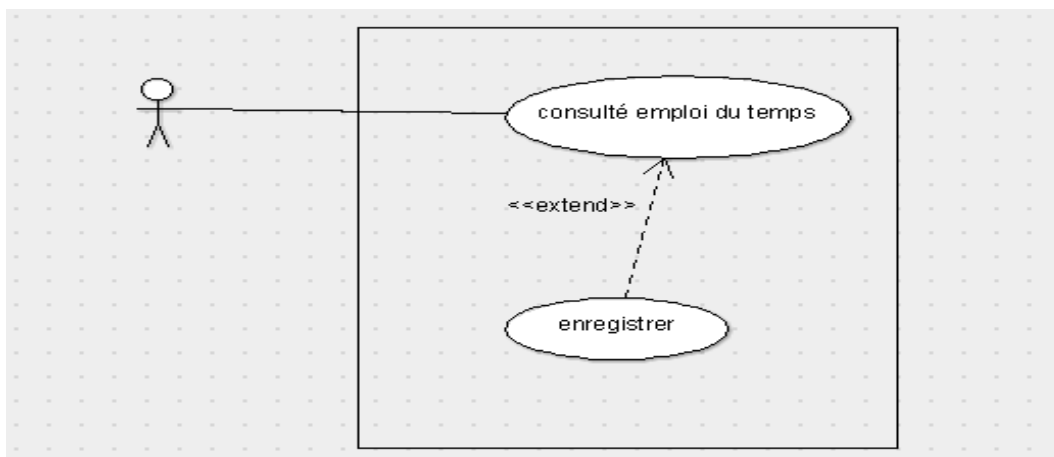


FIGURE 2.6: Diagramme de cas d'utilisation Enregistrer

2.10 Diagramme de séquence Enregistrer

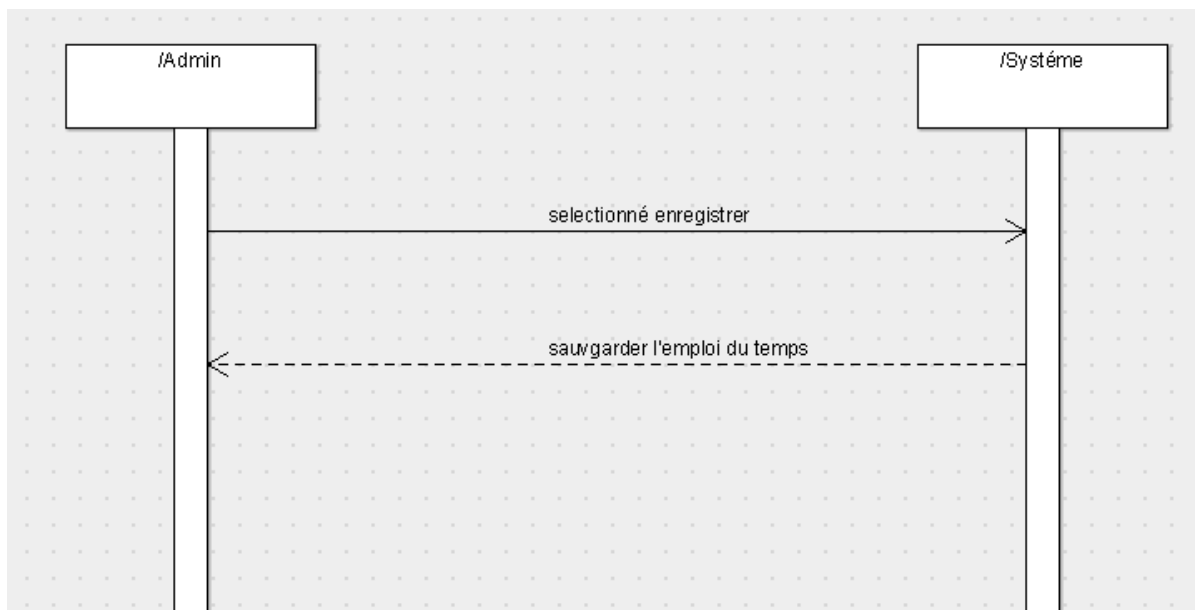


FIGURE 2.7: *Diagramme de séquence Enregistrer*

Description du sommaire	
Titre :	Enregistrer.
But :	Ce cas d'utilisation permet à un administrateur d'enregistrer l'emploi du temps que nous avons créé.
Acteurs :	administrateur.
Description du sommaire	
Enchainements :	1. l'administrateur sélectionne enregistrer ; 2. le système sauvegarde l'emploi du temps.

TABLE 2.3: *Descriptions des cas d'utilisation Enregistrer.*

2.11 Diagramme de cas d'utilisation :Enregistrer sous

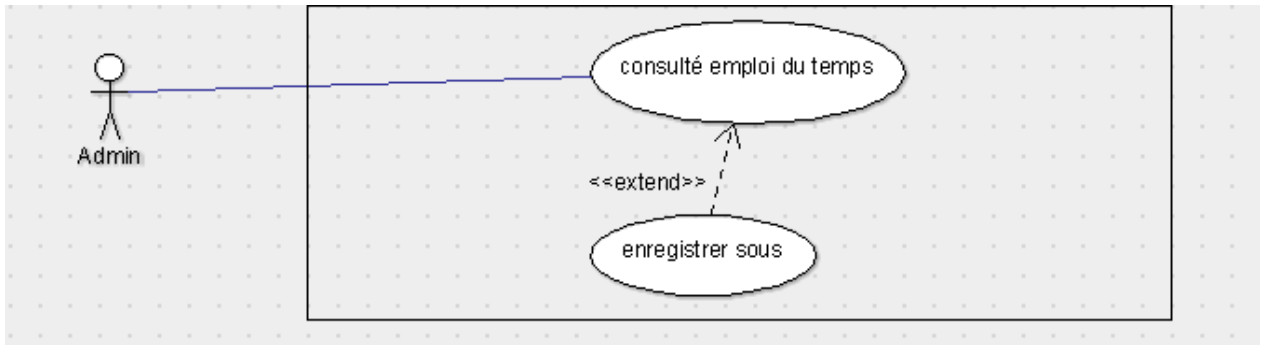


FIGURE 2.8: *Diagramme de cas d'utilisation :Enregistrer sous*

2.12 Diagramme de séquence Enregistrer sous

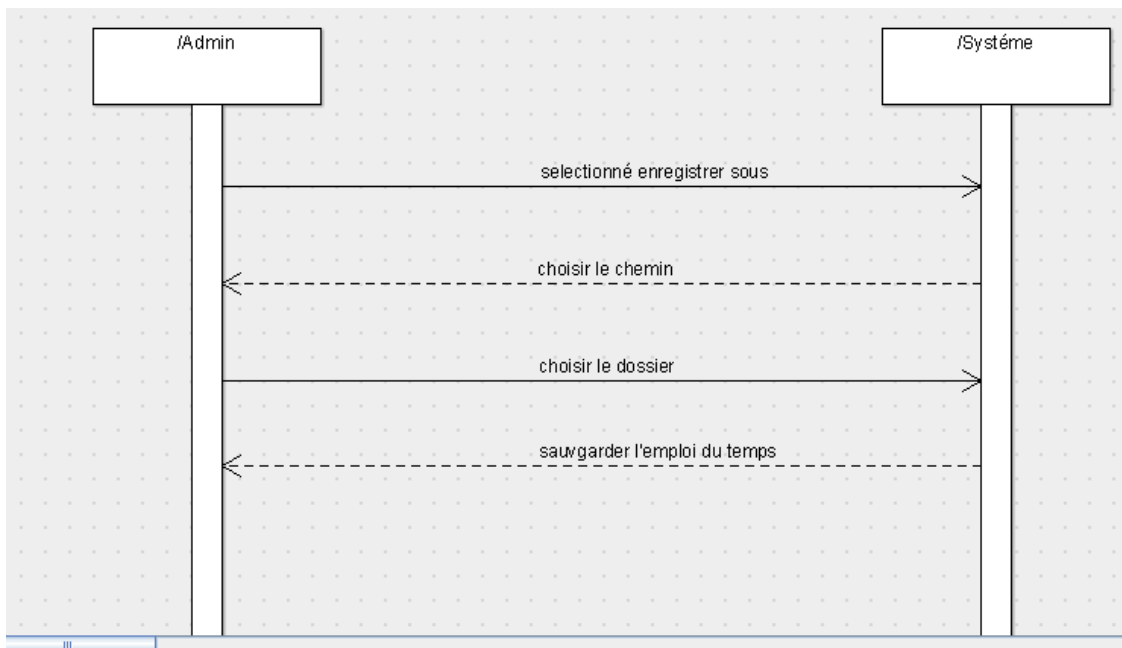


FIGURE 2.9: *Diagramme de séquence Enregistrer sous*

Description du sommaire	
Titre ;	Enregistré sous.
But :	Ce cas d'utilisation permet à un administrateur d'enregistrer l'emploi du temps après avoir choisie un chemin de sauvegarde.
Acteurs :	administrateur.
Description du sommaire	
Enchainements :	<ol style="list-style-type: none"> 1. l'administrateur demande de sauvegarder l'ET avec la sélection de l'ordre "enregistrer sous" ; 2. le système demande de choisir le chemin ; 3. l'administrateur choisie le chemin ; 4. le système enregistre l'ET.

TABLE 2.4: Descriptions de cas d'utilisation Enregistrer sous.

2.13 Diagramme de cas d'utilisation : Imprimer

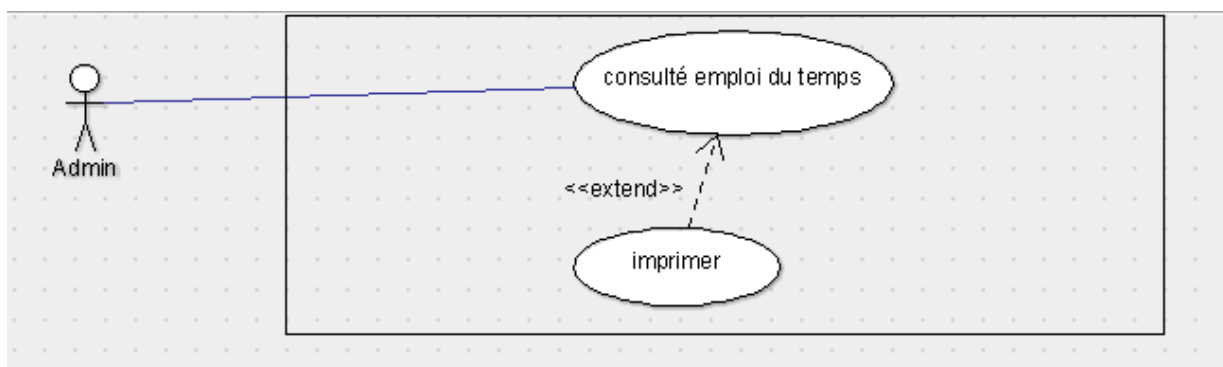


FIGURE 2.10: Diagramme de cas d'utilisation : Imprimer

2.14 Diagramme de séquence Imprimer

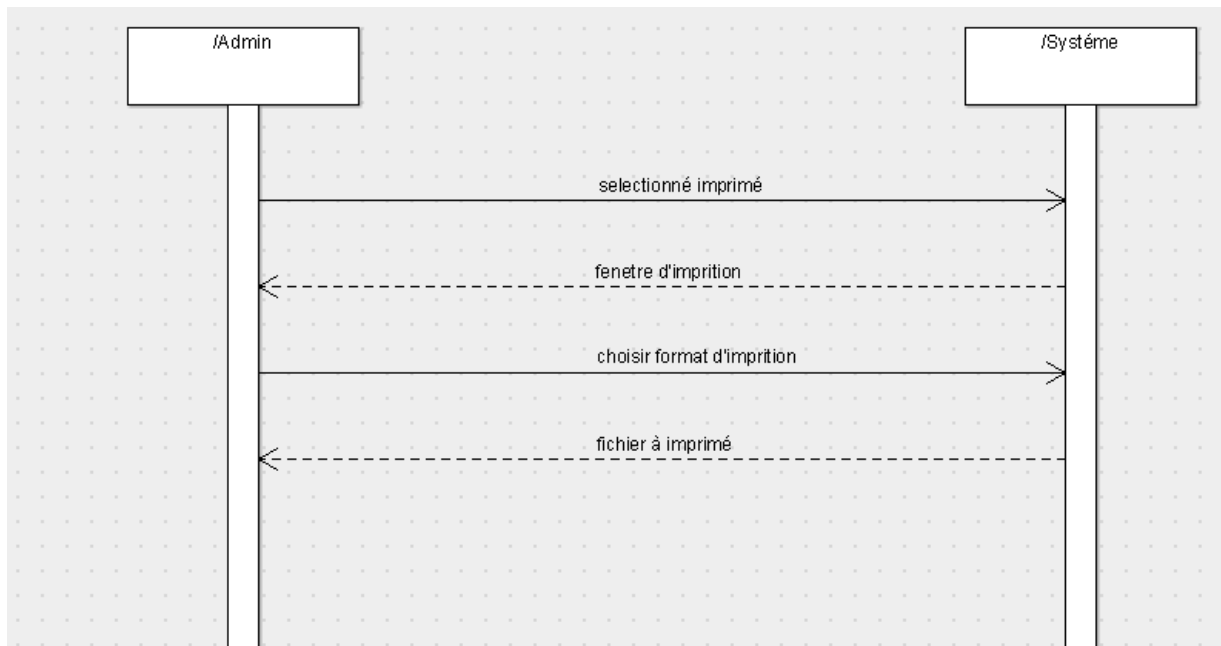


FIGURE 2.11: *Diagramme de séquence Imprimer*

Description du sommaire	
Titre :	Imprimer.
But :	Ce cas d'utilisation permet à un administrateur d'imprimer l' ET .
Acteurs :	administrateur.
Description du sommaire	
Enchainements :	<ol style="list-style-type: none"> 1. l'administrateur demande d'imprimer l'ET ; 2. Le système ouvre une fenêtre d'impression ; 3. l'administrateur choisie la format d'impression ; 4. le système envoie un message "fichier à imprimer".

TABLE 2.5: *Descriptions de cas d'utilisation Imprimer.*

2.15 Diagramme de classe

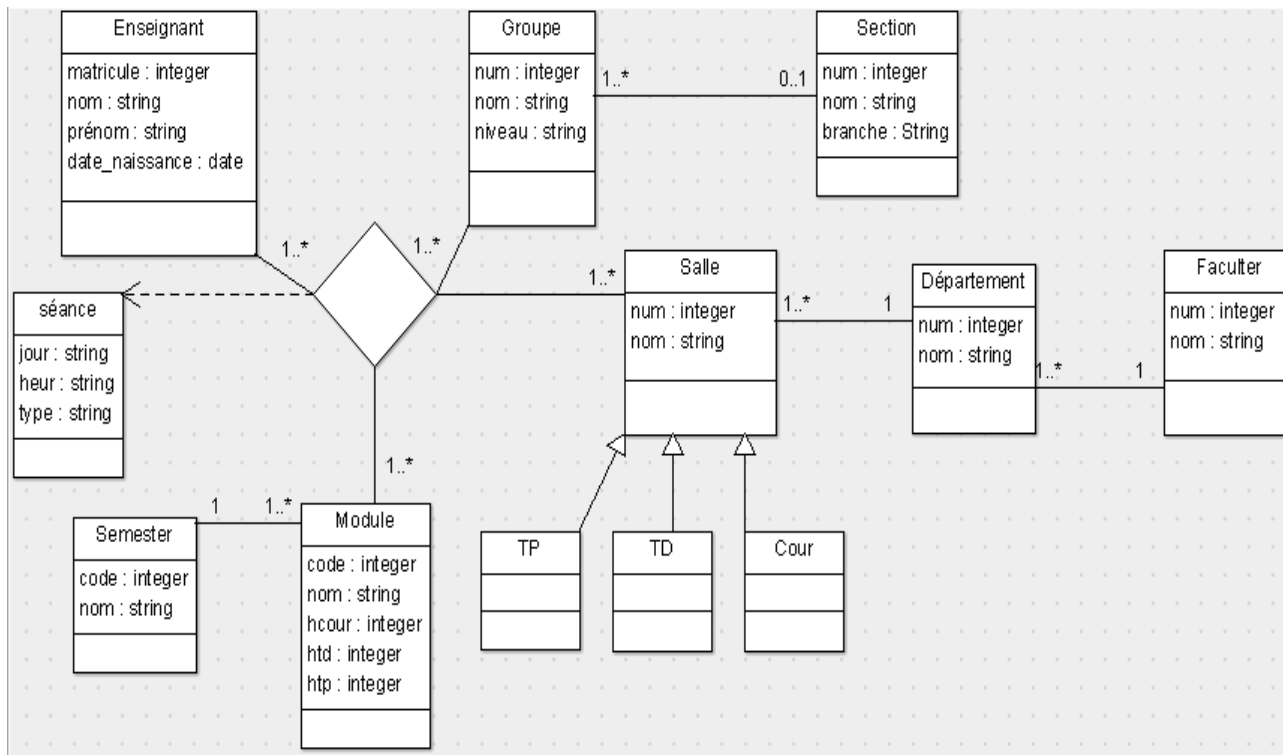


FIGURE 2.12: Diagramme de classe

2.16 Modèle relationnel(MLD)

```

faculter( #num , nom );
departement( #num , nom , numf );
cour ( #num , nom , numDc );
td ( #num , nom , numDtd );
tp ( #num , nom , numDtp );
semestre ( #code , nom );
module( #code , nom , nums , hcour , htd , htp );
section ( #num , nom , branche );
groupe ( #num , nom , numsc , niveau );
enseignant ( #num , nom , prenom , date_naissance );
seance( numE , numM , numG , numTD , numTP , numC , heure , jour , type );

```

2.17 Dictionnaire des données

2.17.1 définition

Un dictionnaire des données est une collection de méta-données ou de données de référence nécessaire à la conception d'une base de données relationnelle. Il revêt une importance stratégique particulière, car il est le vocabulaire commun de l'organisation. Il décrit des données aussi importantes que les clients, les nomenclatures de produits et de services, les annuaires, etc. C'est donc le référentiel principal de l'entreprise, sur lequel s'appuient les décisions de celle-ci. Il est souvent représenté par un tableau à quatre colonnes contenant le nom, le code et le type de donnée ainsi que des commentaires. [6]

Un dictionnaire des données doit respecter les contraintes suivantes :

- Tous les noms doivent être monovalués et non décomposables.
- Il ne doit pas y avoir d'homonymes, ni de synonymes.
- Les données y sont regroupées par entité.
- Les identifiants sont complètement précisés.
- Les commentaires doivent être pertinentes.

Nom	Describe	Domaine	Commentaire	Contrainte
faculter				
num	id_faculté	int	n-auto	not null P.K
nom	nom_faculté	string		not_null <100
Département				
num	id_département	int	n-auto	not null P.K
nom	nom_département	string		not_null <100
numf	id_faculté	int		not_null F.K
cour				
num	id_salle_cours	int	n-auto	not null P.K
nom	nom_salle_cours	string		not_null <20
numDc	id_département	int		not_null F.K
td				
num	id_salle_td	int	n-auto	not null P.K
nom	nom_salle _td	string		not_null <20
numDtd	id_département	int		not_null F.K
tp				
num	id_salle_tp	int	n-auto	not null P.K
nom	nom_salle _tp	string		not_null <20
numDtp	id_département	int		not_null F.K
semestre				
code	id_semestre	int	n-auto	not null P.K
nom	nom_semestre	string		not_null <50
module				
code	id_module	int		not null P.K
nom	nom_module	string		not_null <50
nums	id_semestre	int		not_null F.K
hcour	nbr_cour_semain	int		null
htd	nbr_td_semain	int		null
htp	nbr_tp_semain	int		null

section

num	id_section	int	n-auto	not null P.K
nom	nom_section	string		not_null <10
branche	nom_branche	string		null <50

groupe

num	id_groupe	int	n-auto	not null P.K
nom	nom_groupe	string		not_null <50
numsc	id_section	int		not_null F.K
niveau	nom_niveau	string		not_null <20

enseignant

num	id_enseignant	int		not null P.K
nom	nom_enseignant	string		not_null <50
prenom	prenom_enseignant	string		not_null <50
date_naissance	date_naissance_enseignant	date		not_null

seance

numE	id_enseignant	int		not_null F.K
numM	id_module	int		not_null F.K
numG	id_groupe	int		not_null F.K
numTD	id_salle_td	int		not_null F.K
numTP	id_salle_tp	int		not_null F.K
numC	id_salle_cours	int		not_null F.K
heur	heur_séance	string		not_null <20
jour	jour_séance	string		not_null <10
type	type_séance	string		not_null <10

TABLE 2.6: Dictionnaire de donnée

Chapitre 3

Réalisation et aspect d'implémentation

3.1 Introduction

Dans ce chapitre on implémente notre application en utilisant le langage de programmation **java** avec l'IDE **NetBeans** et **MYSQL** pour la gestion de la base de données et le langage de description **LATEX** pour le traitement de texte.

3.2 l'environnement de programmation JAVA

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitations (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur.[4]

3.3 l'environnement de gestion de BDD MYSQL

MySQL est une base de données relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment).

Le principe d'une base de données relationnelle est d'enregistrer les informations dans des tables, qui représentent des regroupements de données par sujets (table des clients, table des fournisseurs, table des produits, par exemple). Les tables sont reliées entre elles par des relations.

Le langage **SQL** (acronyme de Structured Query Language) est un langage universellement reconnu par MySQL et les autres bases de données et permettant d'interroger et de modifier le contenu d'une base de données. Les autres bases de données utilisées en informatique sont essentiellement Microsoft SQL Server et Oracle.[5]

3.4 L'environnement de traitement de texte LATEX

3.4.1 Définition de LATEX

LATEX est un logiciel dédié à la production de documents professionnels : livres,mémoires,rapportes,articles,...
et un langage de description de documents avec un compilateur associé.

3.4.2 Quelques caractéristiques :

- portabilité.
- interopérabilité.
- libre et gratuit.

3.5 Présentation l'application



FIGURE 3.1: *Interface de connection au système*

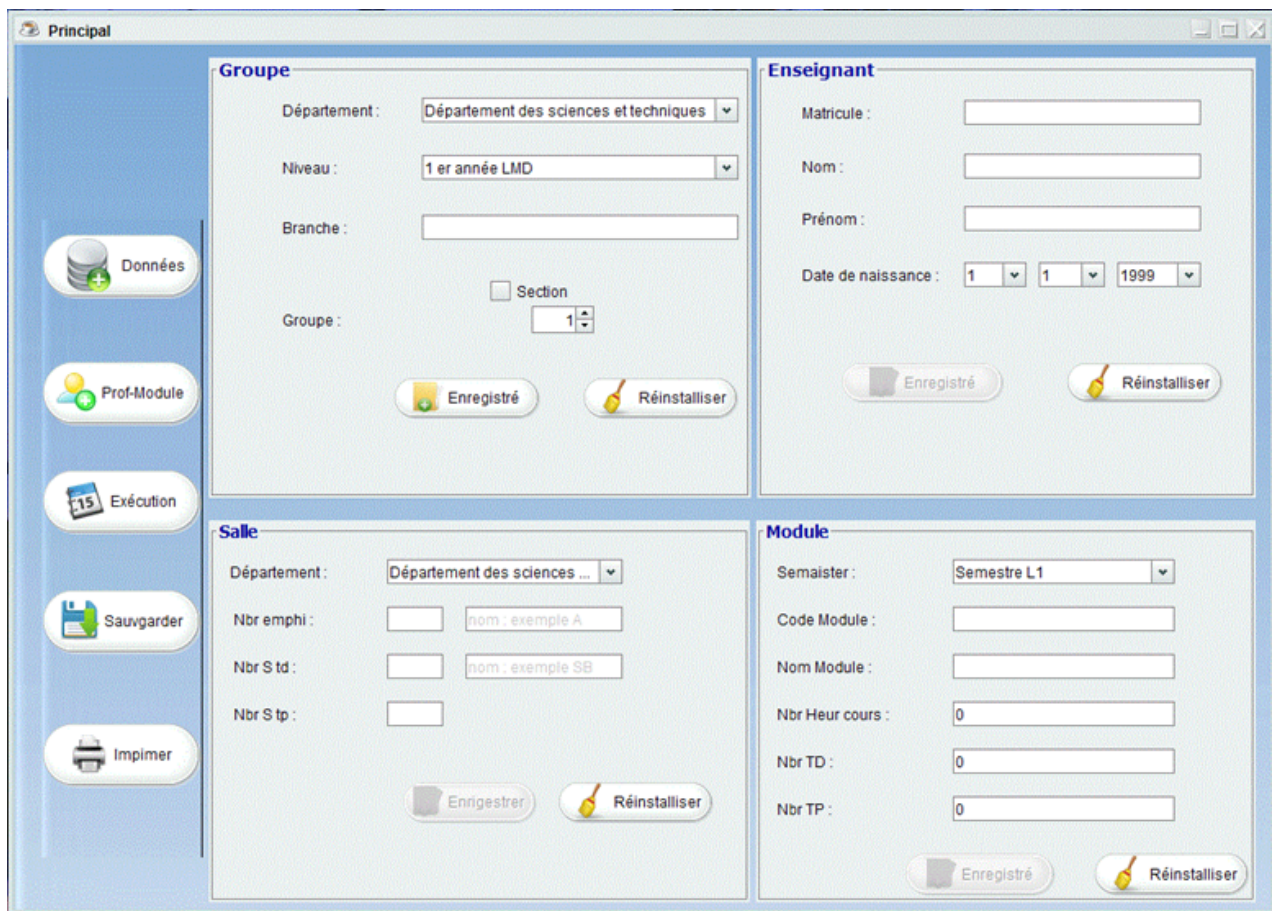


FIGURE 3.2: *Interface principale*

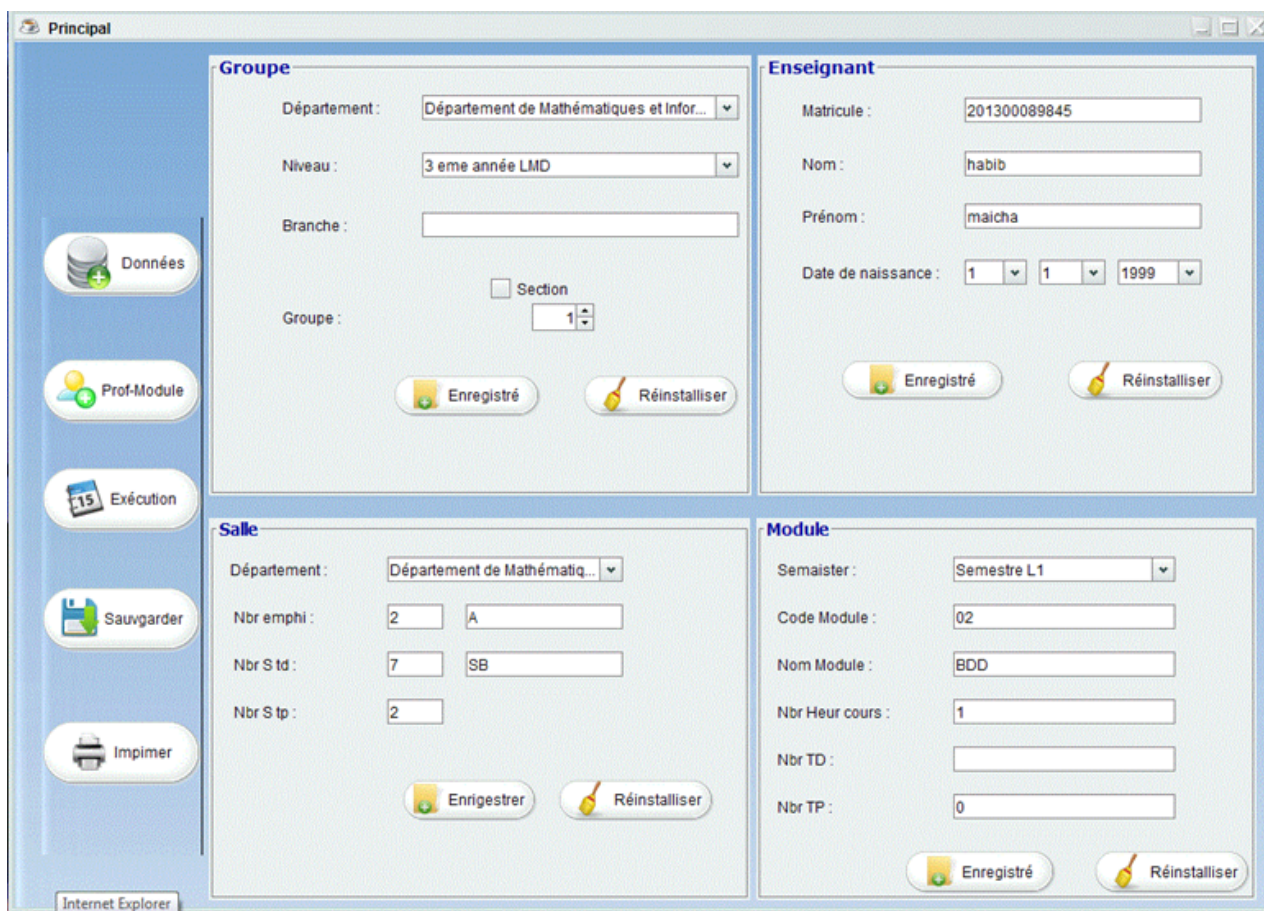


FIGURE 3.3: Interface principale après la saisie

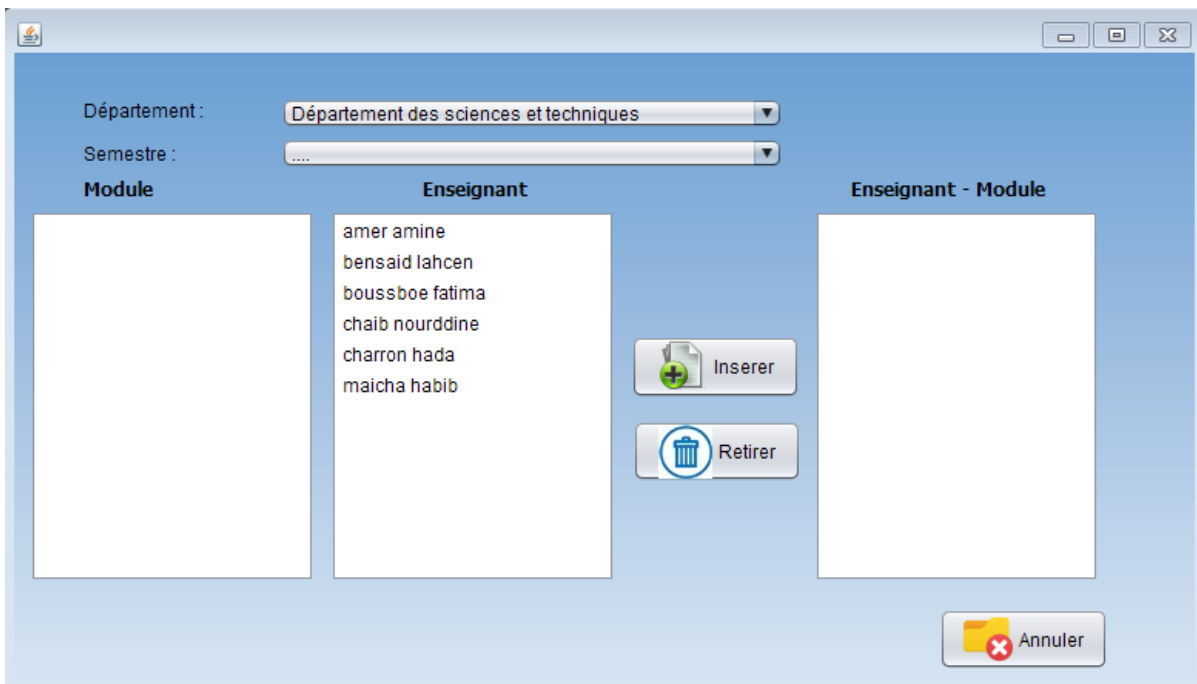


FIGURE 3.4: *Interface prof_module*



FIGURE 3.5: *Interface prof_module après insertion*



FIGURE 3.6: Interface prof_module après la retirer



FIGURE 3.7: Interface de sauvegarde

Conclusion Générale

le problème d'**ET** est très difficile à résoudre. La difficulté réside dans le fait de prendre en compte la grosse masse d'informations en outre des diverses contraintes, particulièrement dans l'établissement scolaire.

le travail qui nous à été présenté consiste à réalisé un système de gestion du temps.

Notre réalisation consiste à programmer un emploi du temps en évitant le conflit.

L'analyse, la spécification, la conception sont décrit en **UML** et le traitement de texte avec le langage de description **LATEX**.

L'implémentation de l'application à été réalisé par l'environnement de programmation **JAVA**.

Bibliographie

- [2] Joseph Gabay David Gabay, UML 2 ANALYSE ET CONCEPTION

webographie

- [1] <https://fr.wikipedia.org/wiki/Probl>
- [3] <http://www.lsis.org/espinasseb/Supports/MR-TC/IntroULM-sept09-4p.pdf>
- [4] <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-java-485/>
- [5] <http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimedia-informatique/m-glossaire-informatique-et-multimedia/448-mysql-definition.html>
- [6] https://fr.wikipedia.org/wiki/Dictionnaire_des_donn