

Département d'Informatique

Licence Sciences et Techniques en Informatique

Option : Génie Logiciel

Mémoire de Projet de Fin d'Etudes

Intitulé :

La gestion des examens à l'aide des algorithmes de graphes : Cas de la FSTE

Préparé par :

- Abdelhadi BAYECHOU

- Mustapha JARIDI

Soutenu le 16 Juillet 2019 devant le Jury :

- Pr. Brahim AKSASSE

- Pr. Aziz BAATAOUI

- Pr. Youssef QARAAI (Encadrant)

Année Universitaire : **2018 / 2019**

Remerciements

Nous tenons tout d'abord à remercier Dieu de nous avoir donné le courage et la patience d'accomplir ce travail.

Nos vifs remerciements accompagnés de toute notre gratitude vont également à notre professeur encadrant M. Youssef QARAAI, pour nous avoir proposé ce sujet et nous avoir guidé pour réaliser ce projet. Les conseils qu'il nous a prodigué, la patience, la confiance qu'il nous a témoigné ont été déterminants dans la réalisation de notre travail.

Nos remerciements vont également aux membres de jury d'avoir accepté d'examiner notre travail.

En fin nous remercions tous ceux qui, de près ou de loin, ont contribué, par leurs conseils, leurs encouragements ou leurs amitiés à l'aboutissement de ce travail.

Merci à tous.

Dédicace

A nos très chers parents,

Aucun mot ne pourra exprimer notre amour envers vous.

A nos très chers frères et sœurs,

Merci pour votre amour inconditionnel.

A nos familles,

A tous nos amis,

Merci pour votre encouragement

A tous nos collègues de la FSTE,

Nous vous souhaitons un grand succès.

A tous ceux qui nous ont soutenus de près ou de loin,

On dédie ce travail.

Abdelhadi BAYECHOU

Mustapha JARIDI

Liste des figures

Figure 1 : Graphe non orienté G1	2
Figure 2 : Graphe complet d'ordre 4	3
Figure 3 : Graphe orienté.....	4
Figure 4 : Arbre et forêt.....	5
Figure 5 : Graphe non orienté G2.....	5
Figure 6 : Graphe complet d'ordre 5.....	6
Figure 7 : Graphe non orienté	9
Figure 8 : Ensemble de couleurs	9
Figure 9 : Itération 1	9
Figure 10 : Itération 2.....	10
Figure 11 : Itération 3.....	10
Figure 12 : Itération 4.....	11
Figure 13 : Itération 5.....	11
Figure 14 : Itération 6.....	11
Figure 15 : Itération 7.....	12
Figure 16 : Les principaux diagrammes UML	14
Figure 17 : Diagramme de cas d'utilisation	16
Figure 18 : Diagramme de séquence afficher planning.....	17
Figure 19 : Diagramme de séquence listes des étudiants	18
Figure 20 : Diagramme de classes.....	19
Figure 21 : Graphe d'incompatibilité	22
Figure 22 : Couleurs utilisés.....	22
Figure 23 : Résultat de la coloration	22
Figure 24 : Principe général pour accéder à une base de données	25
Figure 25 : Interface de l'application	32
Figure 26 : Choix du semestre.....	32
Figure 27 : Choix de la date du début	33
Figure 28 : Date invalide	33
Figure 29 : Choix du type planning.....	33
Figure 30 : Planning enseignants	34
Figure 31 : Planning étudiants.....	35
Figure 32 : Choix du jour et créneau	36
Figure 33 : Choix du local.....	36
Figure 34 : Jour, créneau ou local non choisi.....	36
Figure 35 : Sélection du jour, créneau et local	37
Figure 36 : Liste des étudiants.....	37

Liste des tableaux

Tableau 1 : Exemple de la répartition d'une période de l'examen de la session d'automne	13
Tableau 2 : Echantillon de modules	20
Tableau 3 : Modules et leurs adjacences	21
Tableau 4 : Les principales composantes graphiques	31

Liste des abréviations

UML Unified Modeling Language ou langage de modélisation unifié

FSTE Faculté des Sciences et Techniques Errachidia.

MIP Mathématiques Informatique Physique.

BCG Biologie Chimie Géologie.

DEUST Diplôme d'Etudes Universitaire en Sciences et Techniques.

LST Licence en Sciences et Techniques.

MA Mathématiques Appliquées.

GL Génie Logiciel.

SI Sciences d'Ingénieur.

ER Energies Renouvelables.

BVA Biologie Végétale Appliquée.

PS Physiologie et Santé.

GA Géoscience Appliquée.

CA Chimie Appliquée.

Sommaire

Liste des figures	iii
Liste des tableaux	iv
Liste des abréviations	iv
Introduction générale.....	1
Chapitre I : Coloration de graphes	2
I.1. Généralité	2
I.1.1. Graphes non orientés	2
I.1.2. Graphes orientés	3
I.1.3. Connexité.....	4
I.1.4. Arbres et forêts	4
I.1.5. Matrice d'adjacence.....	5
I.2. Coloration de graphes.....	6
I.2.1. Définitions	6
I.2.2. Algorithme de Welsh and Powell	7
I.2.3. Algorithme de DSATUR	8
I.3. Conclusion.....	12
Chapitre II : Modélisation et conception	13
II.1. Position du problème	13
II.2. Conception	13
II.2.1. Structure des examens au sein de la FSTE.....	13
II.2.2. Modélisation à l'aide du langage UML.....	14
II.3. Coloration et gestion des examens.....	19
II.3.1. Graphe d'incompatibilité.....	20
II.3.2. Exemple.....	20
II.4. Conclusion	23

Chapitre III : Application à la gestion des examens.....	24
III.1. Problématique	24
III.2. Extraction de données.....	24
III.2.1. Outils utilisés	24
III.2.2. Connexion à une base de données MySQL	25
III.3. Noyau de l'application.....	26
III.3.1. Outils de développements	26
III.3.2. Représentation des classes et leurs rôles.....	27
III.4. Interface graphique	30
III.4.1. Moyens utilisés	30
III.4.2. Apparence générale de l'interface.....	32
III.4.3. Affichage du planning.....	32
III.4.4. Affichage des listes des étudiants	36
III.5. Interprétation des résultats.....	38
Conclusion générale	40
Bibliographie	41
Webographie	41

Introduction générale

Dans le processus éducatif, la gestion des examens est l'une des tâches les plus délicates qui doit être planifiée avec une grande précision. Elle exige l'intervention de plusieurs acteurs et le respect de certaines contraintes afin d'éviter tout chevauchement en optimisant la durée et en exploitant au maximum les ressources disponibles.

La gestion des examens dans les établissements d'enseignement, plus particulièrement l'enseignement supérieur, est parmi les grands problèmes rencontrés par les organisateurs. Vu l'augmentation du nombre de modules, vu les réformes pédagogiques qui permettent aux étudiants de suivre un certain nombre de modules, et vu les limitations en ressources humaines et matérielles dans plusieurs établissements, l'organisation des examens devient plus complexe.

L'objectif est de concevoir une solution informatique à la gestion des examens sous forme d'une application bureau pour :

- Optimiser la durée.
- Distribuer les examens sur le plan spatio-temporel.
- Affecter les surveillants et les représentants d'administration aux examens.
- Afficher le planning.
- Afficher les listes des étudiants inscrits dans un examen donné.

Nous présentons, dans ce rapport, la démarche entreprise et les résultats obtenus : dans le premier chapitre nous rappelons quelques concepts théoriques dans le domaine de la théorie des graphes qu'on a utilisé dans le projet. Le second chapitre sera consacré à la modélisation et la conception de notre application (aspect fonctionnelle et aspect structurelle) en abordant la spécification des besoins d'après les cas d'utilisation (identification des acteurs et des cas d'utilisation, relations entre les cas d'utilisation), puis nous présentons la spécification détaillée des besoins en élaborant, pour chaque cas d'utilisation, un diagramme de séquence. Après nous établissons une description structurelle du système à réaliser. En particulier, nous passons à une conception objet en réalisant des classes d'analyse (identification des concepts du domaine, identification des associations et des attributs, élaboration d'un diagramme de classes participantes), et terminons ce chapitre par une modélisation du problème de l'organisation des examens sous forme d'un problème de coloration. Dans le dernier chapitre, nous allons créer une interface graphique pour permettre aux administrateurs de générer les documents qui contiennent les informations nécessaires pour le bon déroulement des examens tel que le planning et les listes des étudiants qui vont passer un examen donné dans un local et horaire bien défini.

Chapitre I : Coloration de graphes

I.1. Généralité

I.1.1. Graphes non orientés

I.1.1.a. Définition

Un graphe G est un couple formé de deux ensembles : un ensemble non vide et fini $X = \{x_1, x_2, \dots, x_n\}$ dont les éléments sont appelés sommets, et un ensemble fini $A = \{a_1, a_2, \dots, a_m\}$ dont les éléments sont appelés des arêtes [1].

On dit que a est une arête de G d'extrémités x et y lorsque $a = \{x, y\} \in A$ avec $x, y \in X$, et on note $G = (X, A)$.

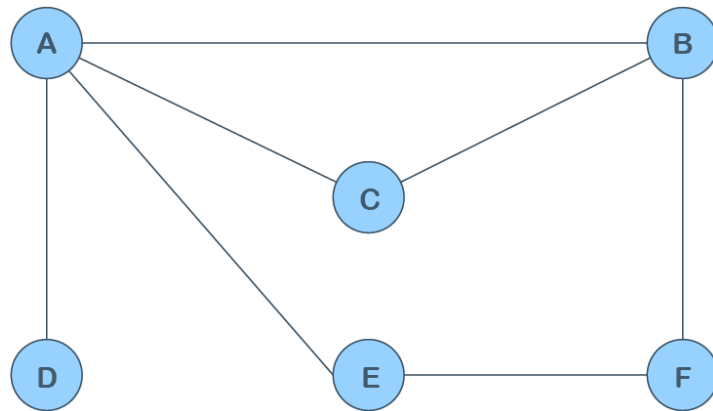


Figure 1 : Graphe non orienté $G1$

- Remarques

- Un graphe d'ordre n est un graphe ayant n sommets.
- $G2$ est un sous-graphe de $G1$ si et seulement si $G2$ est composé de certains sommets de $G1$ et de toutes les arêtes qui les relient dans $G1$.

I.1.1.b. Exemples

- Graphe d'un Tournoi $T = (X, A)$ avec :
 - X : Equipes participantes au tournoi.
 - A : Ensemble des paires d'équipes qui se rencontrent au tournoi.
- La carte routière d'un certain pays $P = (X, A)$ avec :
 - X : Les villes de ce pays.
 - A : Le trajet entre deux villes données.

Cas particulier

On appelle graphe complet, un graphe dont l'ensemble de ses sommets sont reliés par une arête deux à deux et dont le nombre de ses arêtes vaut $\frac{n(n-1)}{2}$ avec n le cardinal de l'ensemble des sommets X .

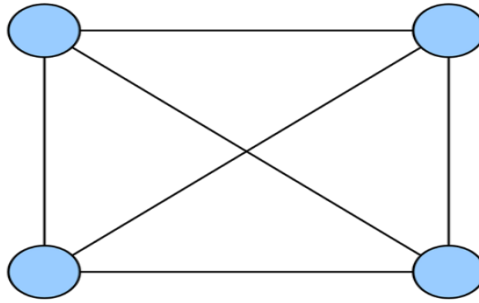


Figure 2 : Graphe complet d'ordre 4

I.1.1.c. Degré dans un graphe non orienté

Soit $G = (X, A)$ un graphe, et x un sommet de ce graphe. Le degré de x , noté $d(x)$, est le nombre des arêtes incidentes à x , c'est-à-dire contenant x .

Lorsque $d(x) = 0$, on dit que x est isolé.

- **Exemple** : le graphe de la figure 1.

Le degré du sommet A est : $d(A) = 4$.

Le degré du sommet B est : $d(B) = 3$.

- **Lemme de poignées de main**

Dans un graphe non orienté $G = (X, A)$, on a :

$$\sum_{x \in X} d(x) = 2|A|$$

I.1.2. Graphes orientés**I.1.2.a. Définition**

Un graphe orienté G est formé de deux ensembles : un ensemble $X = \{x_1, x_2, \dots, x_n\}$ non vide et fini dont les éléments sont appelés sommets, et un ensemble $A = \{a_1, a_2, \dots, a_m\}$ dont les éléments sont appelés des arcs. On notera $G = (X, A)$ [1].

Si $a = (x, y)$ est un arc du graphe G , x est l'extrémité initiale de a et y l'extrémité finale de a .

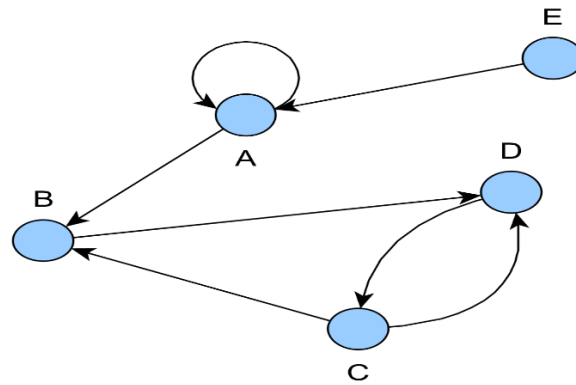


Figure 3 : Graphe orienté

I.1.2.b. Degré d'un graphe orienté

-Définition

Soit x un sommet d'un graphe orienté $G=(X,A)$. On note $d^+(x)$ le nombre d'arcs ayant x comme extrémité initiale (degré sortant), et $d^-(x)$ le nombre d'arcs ayant x comme extrémité finale (degré entrant).

-Propriétés

- $d(x)=d^+(x)+d^-(x)$.
- $\sum_{x \in X} d^+(x) = \sum_{x \in X} d^-(x)$.

I.1.3. Connexité

I.1.3.a. Définitions

- Soit $G=(X,A)$ un graphe non orienté. On appelle chaîne, une séquence d'arêtes telles que chaque arête ait une extrémité commune avec la suivante.
- Une chaîne est un cycle si la première et la dernière extrémité coïncident.
- Un Chemin est une suite des sommets reliés par des arcs dans un graphe orienté.
- Un circuit est un chemin qui revient à son point de départ dans un graphe orienté.

I.1.3.b. Graphe connexe

Un graphe non orienté $G=(X,A)$ est dit connexe si : $\forall x_i, x_j \in X$, Il existe une chaîne reliant x_i et x_j .

Dans le cas d'un graphe orienté on dit que $G=(X,A)$ est connexe si : $\forall x_i, x_j \in X$, Il existe un chemin reliant x_i et x_j .

I.1.4. Arbres et forêts

I.1.4.a. Définition

On appelle arbre tout graphe connexe et sans cycle. Un graphe sans cycle mais non connexe est appelé forêt.

Une feuille est un sommet de degré 1.

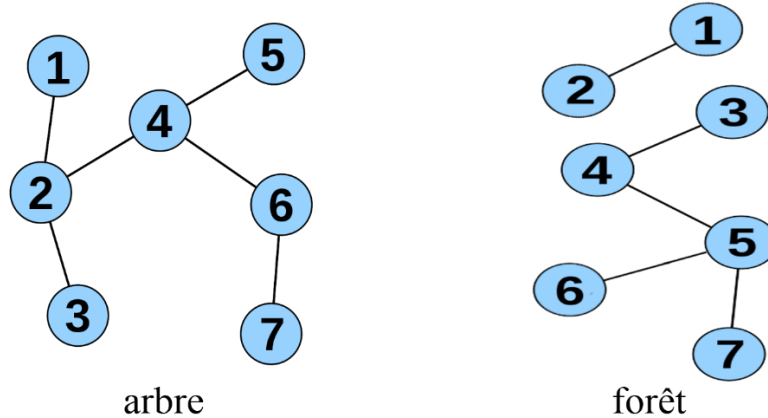


Figure 4 : Arbre et forêt

- Pour l'arbre, les sommets 1,3,5 et 7 sont des feuilles.
- Pour la forêt, les sommets 1,2,3,6 et 7 sont des feuilles.

I.1.4.b. Propriétés

Soit $G=(X,A)$ un graphe d'ordre n . L'arbre est caractérisé par :

- G est connexe et sans cycle.
- G est sans cycle et contient $(n-1)$ arêtes.
- Entre deux sommets quelconques de G , il existe une et une seule chaîne.
- En ajoutant une arête, G devient cyclique.
- En supprimant une arête, G n'est plus connexe.

I.1.5. Matrice d'adjacence

Dans un graphe, on dit que deux sommets sont adjacents s'ils sont liés par une arête.

I.1.5.a. Définition

Soit $G=(X,A)$ un graphe non orienté, avec $X=\{x_1, x_2, \dots, x_n\}$. La matrice d'adjacence du graphe G est la matrice $M(G) \in \mathcal{M}_n(\mathbb{IR})$ dont les coefficients $m_{i,j}$ sont définis par :

$$m_{i,j} = \begin{cases} 1 & \text{si } (x_i, x_j) \in A \\ 0 & \text{si } (x_i, x_j) \notin A \end{cases}$$

I.1.5.b. Exemple

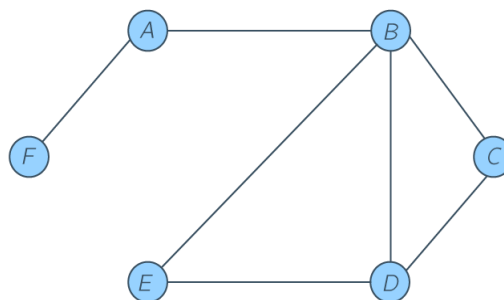


Figure 5 : Graphe non orienté G2

La matrice d'adjacence de G2 :

$$M(G2) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

I.2. Coloration de graphes

I.2.1. Définitions

I.2.1.a. Coloration

La coloration des sommets d'un graphe consiste à affecter à tous les sommets de ce graphe une couleur de telle sorte que deux sommets adjacents ne portent pas la même couleur.

Si le graphe est coloré en k couleurs, on dit qu'on a une k coloration du graphe [3].

I.2.1.b. Nombre chromatique

Le nombre chromatique d'un graphe G noté $\gamma(G)$ [2] est le nombre minimum de couleurs nécessaires à sa coloration, c'est-à-dire le plus petit nombre de couleurs permettant de colorier tous les sommets du graphe sans que deux sommets adjacents soient de la même couleur.

Le nombre chromatique est inférieur ou égale à $r+1$, avec r le plus grand degré du graphe et supérieur ou égale à p , avec p l'ordre du plus grand sous-graphe complet.

$$p \leq \gamma(G) \leq r+1$$

I.2.1.c. Exemple

Soit le graphe complet d'ordre 5 suivant :

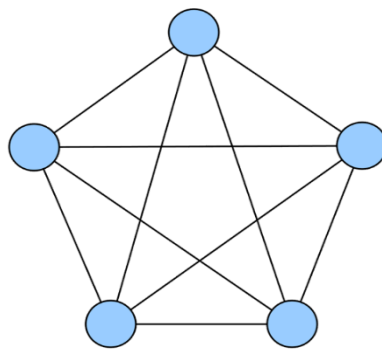


Figure 6 : Graphe complet d'ordre 5

Le nombre chromatique de ce graphe est égale à 5.

En effet, le plus grand degré r du graphe est égale à 4 donc $\gamma(G) \leq r+1$ c'est-à-dire $\gamma(G) \leq 5$ et l'ordre p du plus grand sous-graphe complet est égale à 5 c'est-à-dire $\gamma(G) \geq 5$. Par suite, on a $\gamma(G) \leq 5$ et $\gamma(G) \geq 5$ d'où $\gamma(G) = 5$.

I.2.2. Algorithme de Welsh and Powell

I.2.2.a. Principe

L'algorithme de Welsh and Powell [4] consiste à colorer séquentiellement le graphe en visitant les sommets par ordre de degré décroissant. L'idée est que les sommets ayant beaucoup de voisins seront plus difficiles à colorer, et donc il faut les colorer en premier.

I.2.2.b. Fonctionnement

Il y a une certaine procédure à respecter afin de mettre en œuvre cet algorithme.

-Etape 1

Classer les sommets du graphe dans l'ordre décroissant de leurs degrés et attribuer à chacun des sommets son numéro d'ordre dans la liste obtenue.

On obtient une liste ordonnée de sommets X_1, X_2, \dots, X_n tel que :

$$\text{degré}(X_1) \geq \text{degré}(X_2) \geq \dots \geq \text{degré}(X_n)$$

-Etape 2

En parcourant la liste dans l'ordre, attribuer une couleur non encore utilisée au premier sommet non encore coloré, et attribuer cette même couleur à chaque sommet non encore coloré et non adjacent à un sommet de cette couleur.

-Etape 3

S'il reste des sommets non colorés dans le graphe, revenir à l'étape 2.

Sinon, la coloration est terminée.

I.2.2.c. Extrait de l'algorithme

On note L la liste des sommets s classés selon l'ordre décroissant de leur degré et on note $E(s)$ l'ensemble des sommets adjacents à s .

<i>Couleur = 0</i>
<i>Tant que $L \neq \emptyset$ faire</i>
<i>Couleur = Couleur + 1</i>
<i>Couleur(s) = Couleur</i>
<i>Pour tout t dans L faire</i>
<i>Si $t \notin E(s)$ alors</i>
<i>Couleur(t) = Couleur</i>
<i>$E(s) = E(s) \cup E(t)$</i>
<i>Fin si</i>
<i>Fin pour</i>
<i>Retirer les sommets colorés de L</i>
<i>Fin tant que</i>

I.2.3. Algorithme de DSATUR

I.2.3.a définition

DSATUR ou DSAT [3] (degré de saturation) est un algorithme de coloration de graphes créé par Daniel Brélaz en 1979 à l'EPFL (École Polytechnique Fédérale de Lausanne). Il s'agit d'un algorithme de coloration séquentiel par heuristique.

On a alors l'heuristique appelée DSATUR qui est une des plus connues pour le problème de coloration de graphes.

I.2.3.b. Principe

On considère un graphe $G=(V,E)$ non orienté. Pour chaque sommet v de V , on calcule le degré de saturation $DSAT(v)$ et l'on utilisera ce nombre ainsi que le degré des sommets pour déterminer l'ordre de coloration du graphe. L'algorithme s'arrête lorsque tous les sommets de G sont colorés.

L'algorithme DSATUR est un algorithme de coloration séquentiel, au sens où il colore un seul sommet non déjà coloré à la fois et tel qu'au départ le graphe n'est pas coloré.

I.2.3.c. Fonctionnement

Les différentes étapes sont :

-Etape 1

Ordonner les sommets par ordre décroissant de degrés.

-Etape 2

Colorer le sommet de degré maximum avec la couleur 1.

-Etape 3

Choisir un sommet avec DSAT maximum. En cas d'égalité, choisir un sommet de degré maximal.

-Etape 4

Colorer ce sommet avec la plus petite couleur possible.

-Etape 5

Si tous les sommets sont colorés alors stop. Sinon aller en 3.

I.2.3.d. Extrait de l'algorithme

Soit i un sommet.

Le degré de saturation $DSAT(i)$ est défini de la manière suivante :

Si aucun sommet adjacent à i n'est coloré alors
 $DSAT(i) = \text{degré}(i)$
 Sinon
 $DSAT(i) = \text{nombre de couleurs différentes utilisées par les sommets adjacents à } i$

POUR i allant de 1 à n poser $DSAT(i) = \text{degré}(i)$
 Numéroter les couleurs
 TANTQUE il existe un sommet non coloré
 Choisir un sommet i avec $DSAT(i)$ est maximum
 Si $DSAT(i) = DSAT(j)_{j=1,2, \dots, n}$ alors

Choisir un sommet i avec $d(i)$ est maximum

Si $d(i)=d(j)_{j=1,2, \dots, n}$ alors

Choisir i ou j

Colorier le sommet i avec la première couleur possible

Mettre à jour DSAT pour les sommets adjacents à ce sommet

FINTANTQUE

I.2.3.e. Exemple

Soit le graphe suivant :

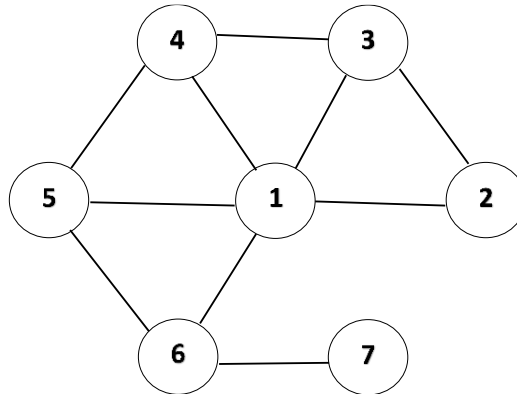


Figure 7 : Graphe non orienté

On désire colorer le graphe ci-dessus moyennant l'algorithme DSATUR avec les couleurs suivantes selon le besoin :



Figure 8 : Ensemble de couleurs

-Itération 1

Aucun sommet n'est coloré, alors on colore le sommet avec le plus grand degré par la couleur C1. C'est le sommet 1.

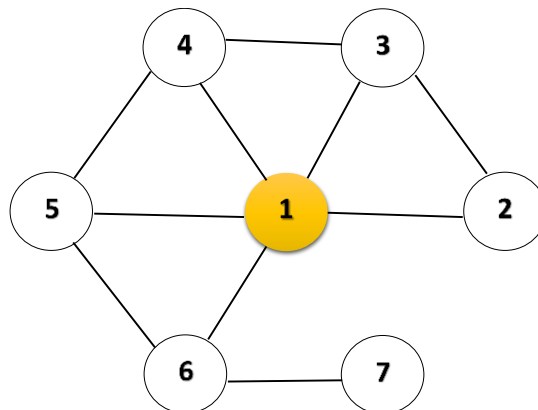
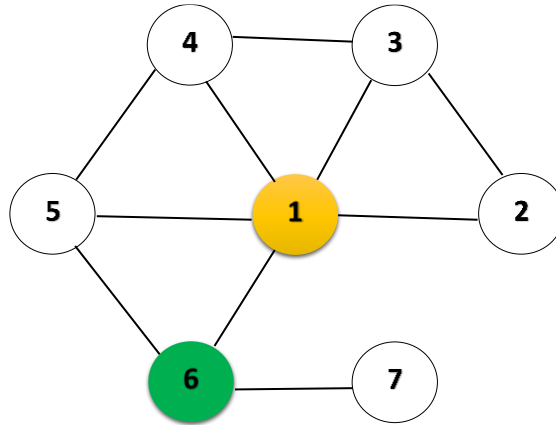


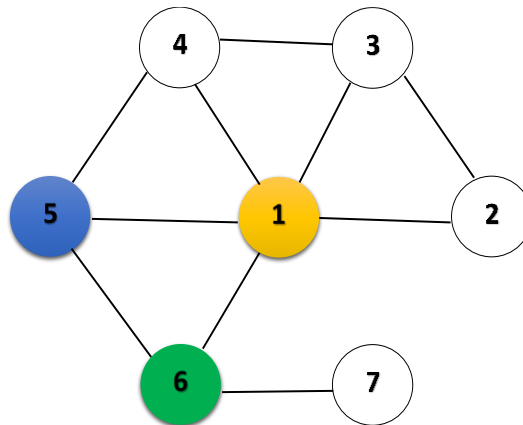
Figure 9 : Itération 1

-Itération 2

On a 5 sommets (2, 3, 4, 5 et 6) ayant 1 pour degré de saturation. Alors, on choisit parmi eux celui qui a le plus grand degré. Ici, on a le choix entre les sommets (3, 4, 5, et 6) car ils ont le même degré ($d = 3$). Nous allons prendre le sommet 6 et lui attribuer la couleur C2 (on ne peut pas lui attribuer la couleur C1 car il est adjacent avec le sommet 1).

*Figure 10 : Itération 2***-Itération 3**

On a le sommet 5 qui est le seul sommet non coloré adjacent à deux sommets colorés, donc il a le plus grand degré de saturation qui vaut 2. Ainsi, on lui attribue la couleur C3 puisqu'il est adjacent aux sommets 1 et 6.

*Figure 11 : Itération 3***-Itération 4**

On a le sommet 4 qui est le seul sommet non coloré adjacent à deux sommets colorés, donc il a le plus grand degré de saturation qui vaut 2. Ainsi, on lui attribue la couleur C2 puisqu'il n'est pas adjacent au sommet 6.

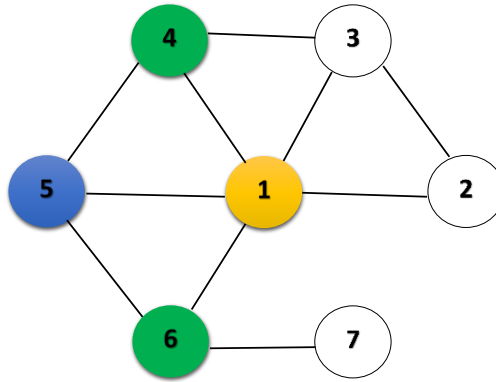


Figure 12 : Itération 4

-Itération 5

Le sommet 3 est le seul sommet non coloré adjacent à deux sommets colorés, donc il a le plus grand degré de saturation qui vaut 2. Ainsi, on lui attribue la couleur C3 puisqu'il n'est pas adjacent au sommet 5.

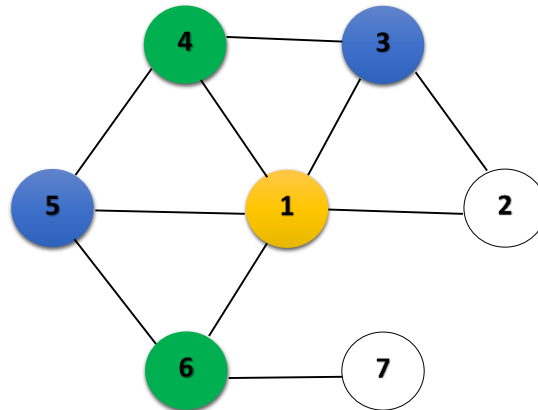


Figure 13 : Itération 5

-Itération 6

On a le sommet 2 qui est le seul sommet non coloré adjacent à deux sommets colorés, donc il a le plus grand degré de saturation qui vaut 2. Ainsi, on lui attribue la couleur C2 puisqu'il n'est pas adjacent au sommets 4 et 6.

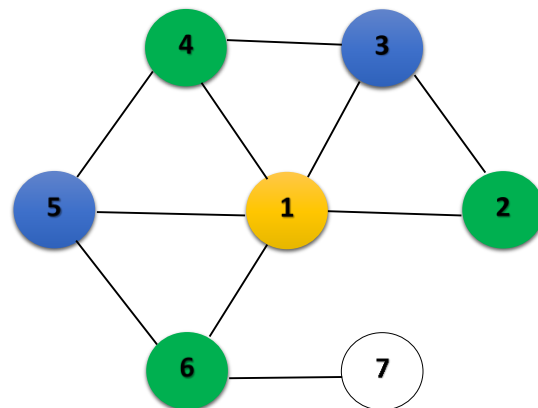


Figure 14 : Itération 6

-Itération 7

Il reste un sommet non coloré, on lui attribue la couleur C1 car il n'est pas adjacent au sommet 1 ou C2 puisqu'il n'est pas adjacent aux sommets 3 et 5.

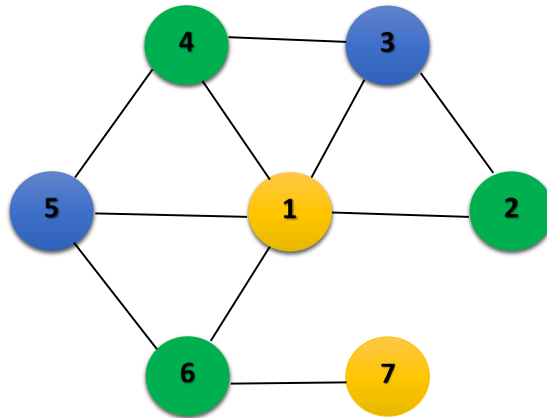


Figure 15 : Itération 7

Ainsi, le graphe a été coloré avec trois couleurs :

- C1 pour les sommets 1 et 7,
- C2 pour les sommets 2, 4 et 6,
- C3 pour les sommets 3 et 5.

I.3. Conclusion

Nous avons présenté dans ce chapitre quelques concepts sur la théorie des graphes qui contribuent dans la solution de plusieurs problèmes dans différents domaines, surtout la coloration des graphes qui constitue une solution à tout problème de graphes dits d'incompatibilité comme par exemple celui qui modélise la planification des emplois du temps ou des examens.

Chapitre II : Modélisation et conception

II.1. Position du problème

La planification des examens au sein de la Faculté des Sciences et Techniques d'Errachidia dépend de plusieurs facteurs tels que :

- Augmentation de l'effectif des inscrits.
- Disponibilité des locaux.
- Ressources humaines : professeurs et administrateurs.
- Problème des chevauchements.
- Réforme pédagogique : nature du système modulaire.

Pendant ces dernières années, on a constaté que la durée réservée au déroulement des examens s'étale sur un nombre remarquable de semaines qui peut atteindre au moyenne 6 semaines allant de la phase de la préparation à l'affichage des résultats finaux, en passant par la période des rattrapages.

Préparation	Session normal	Affichage	Rattrapage
Deux semaines	Deux semaines	Une semaine	Une semaine

Tableau 1 : Exemple de la répartition d'une période de l'examen de la session d'automne

Pour remédier à ce problème, nous avons été chargés de réaliser une application qui a comme objectif d'optimiser au maximum possible la durée du déroulement des examens moyennant l'algorithme de coloration DSATUR et de générer un planning contenant toutes les informations qui définissent un examen (code module, désignation, responsable, section, horaire, local, surveillants et représentant administratif).

II.2. Conception

Il est important, avant de commencer toute application informatique, d'esquisser les concepts théoriques permettant d'asseoir la méthodologie à suivre et les outils à utiliser afin d'arriver à la solution souhaitée par les utilisateurs de cette application.

II.2.1. Structure des examens au sein de la FSTE

Actuellement chaque étudiant inscrit à la FSTE (parcours MIP ou BCG) doit valider 24 modules distribués sur quatre semestres (S1, S2, S3, S4) dont chacun contient 6 modules pour avoir son DEUST.

Après avoir validé au moins 21 modules de tronc commun, il pourra s'inscrire à une des spécialités disponibles selon son parcours et son choix (selon la décision de la commission pédagogique).

A noter, le parcours MIP dispose de quatre spécialités (LSTMA, LSTGL, LSTSI, LSTER) contenant chacune six modules en S5 et trois en S6 sauf LSTSI qui contient trois options en

S6. Et le parcours BCG dispose aussi de quatre spécialités (LSTBVA, LSTCA, LSTPS, LSTGA) contenant chacune six modules en S5 et trois en S6.

La faculté des sciences et techniques d'Errachidia organise deux sessions des examens par an. Pour une session donnée, l'examen d'un module est planifié s'il a un effectif supérieur ou égale à un étudiant.

II.2.2. Modélisation à l'aide du langage UML

II.2.2.a. Définition

UML (Unified Modeling Language, que l'on peut traduire par "langage de modélisation unifié") est une méthode permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, et est devenu désormais la référence en termes de modélisation objet.

UML est un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, à spécifier et documenter des systèmes, à esquisser des architectures logicielles, à concevoir des solutions et communiquer des points de vue.

Cet outil unifie les notations et les concepts orienté objet. Il unifie également les notations nécessaires à la description des activités d'un processus de développement allant de l'expression des besoins jusqu'au codage. Il s'articule autour de 14 diagrammes répartis en 7 diagrammes structurels et 7 diagrammes comportementaux.

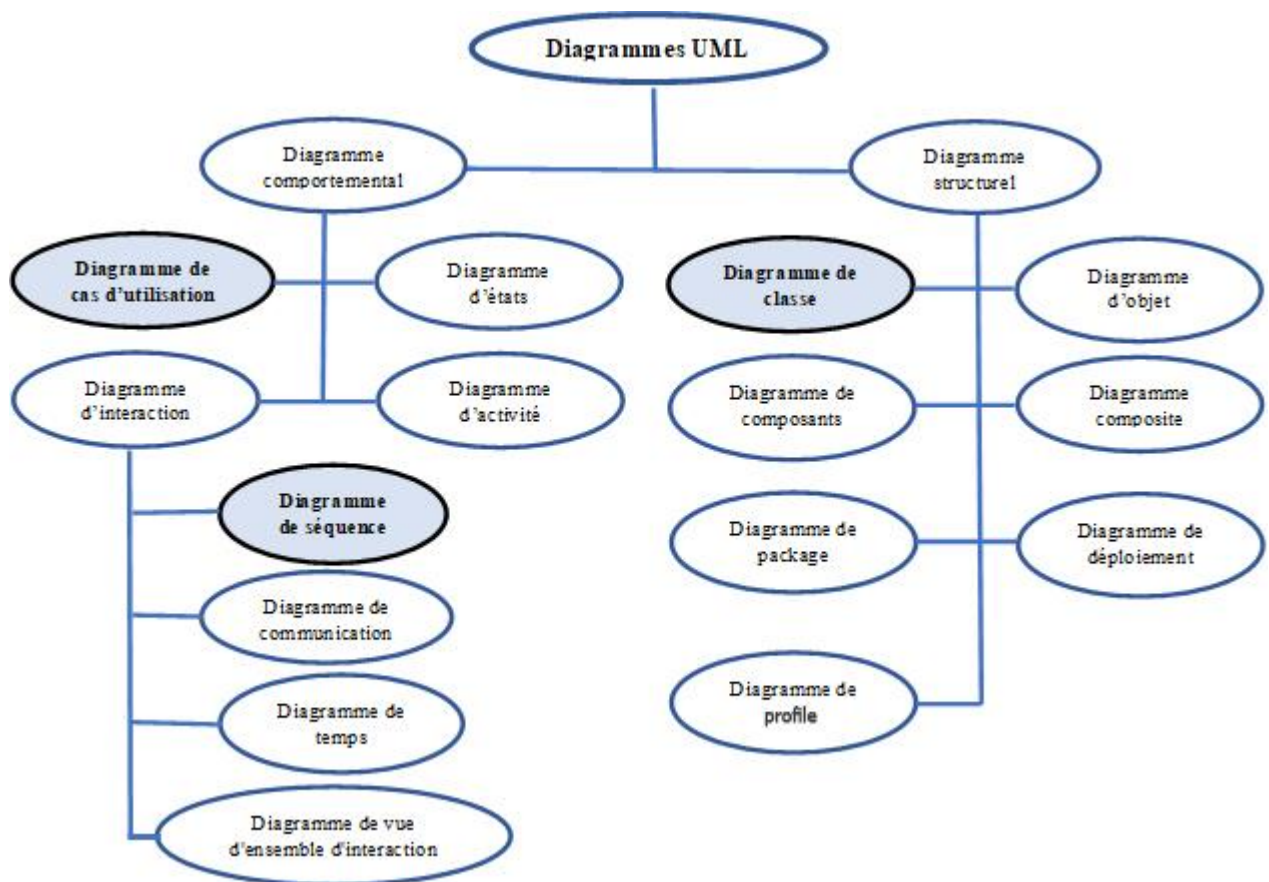


Figure 16 : Les principaux diagrammes UML

II.2.2.b. Outil utilisé : AstahUML



Anciennement appelé Jude, Astah est un outil de modélisation UML créé par la compagnie japonaise ChangeVision1. Il fonctionne avec l'environnement d'exécution Java. Le nom vient de l'acronyme Java and UML developers environment.

Astah est un logiciel propriétaire distribué gratuitement en version 'Community'. L'achat d'une licence "Professional" permet de bénéficier d'un support client. Une fonctionnalité notable du logiciel est l'exportation en Java du modèle UML créé.

Astah supporte officiellement les systèmes Windows, mais peut aussi fonctionner sous Linux et MacOS.

AstahUML permet de créer :

- Des diagrammes des classes.
- Des diagrammes des cas d'utilisation.
- Des diagrammes d'états.
- Des diagrammes d'activités.
- Des diagrammes de collaboration.
- Des diagrammes de déploiement.

II.2.2.c. Diagrammes utilisés

Avant d'entamer la modélisation du système objet d'étude, nous présentons une vue sur les diagrammes utilisés pour cet objectif :

- **Diagramme de cas d'utilisation** : appartient à la partie comportement d'UML. Il consiste à définir un ensemble d'acteurs qui interagissent avec le système en spécifiant à chaque acteur les services fournis (cas d'utilisation) par le système. Ils montrent l'interaction fonctionnelle entre les acteurs sollicitant des services et le système répondant à ces services.
- **Diagramme de séquence** : il est assimilé à la partie du diagramme d'interaction. Il permet de représenter graphiquement les scénarios d'utilisation d'un cas d'utilisation. C'est la représentation graphique des interactions entre les acteurs et le système, selon un ordre chronologique.
- **Diagramme de classe** : il s'agit de la partie statique d'UML. Il décrit l'aspect structurelle de notre projet, c'est-à-dire il décrit les entités contribuant dans le système sous forme des classes. La classe est définie comme étant l'abstraction d'un ensemble d'objets ayant une sémantique et des caractéristiques communes.

II.2.2.d. Diagramme de cas d'utilisation

La première étape pour modéliser un système est de fixer les acteurs concernés, ainsi que les services fournis pour chacun d’eux, c’est ce qu’on appelle la spécification des besoins qui se fait par un diagramme de cas d’utilisation.

- Identification des acteurs

- L'administrateur : le seul acteur principal qui interagit avec le système, il peut afficher le planning des examens et la liste des étudiants relativement à un examen, un horaire et un local.
- La base de données : un acteur secondaire qui se comporte comme une source de données. En effet les données des modules, les étudiants, leurs inscriptions, les professeurs et les administrateurs y sont stockés.

- Identification des cas d'utilisation

- **Afficher planning** : l'administrateur affiche le planning des examens, mais avant il faut fixer un semestre, une date de début et un type de planning (planning enseignants ou planning étudiants), et il faut que le planning soit établi en mettant en jeu les données extraits de la base de données (modules, étudiants inscrits, professeurs et représentants).
- **Afficher listes des étudiants qui vont passer le même examen** : l'administrateur affiche la liste des étudiants relativement à un examen, un horaire et un local, après avoir établi le planning par le système.

- Diagramme de cas d'utilisation

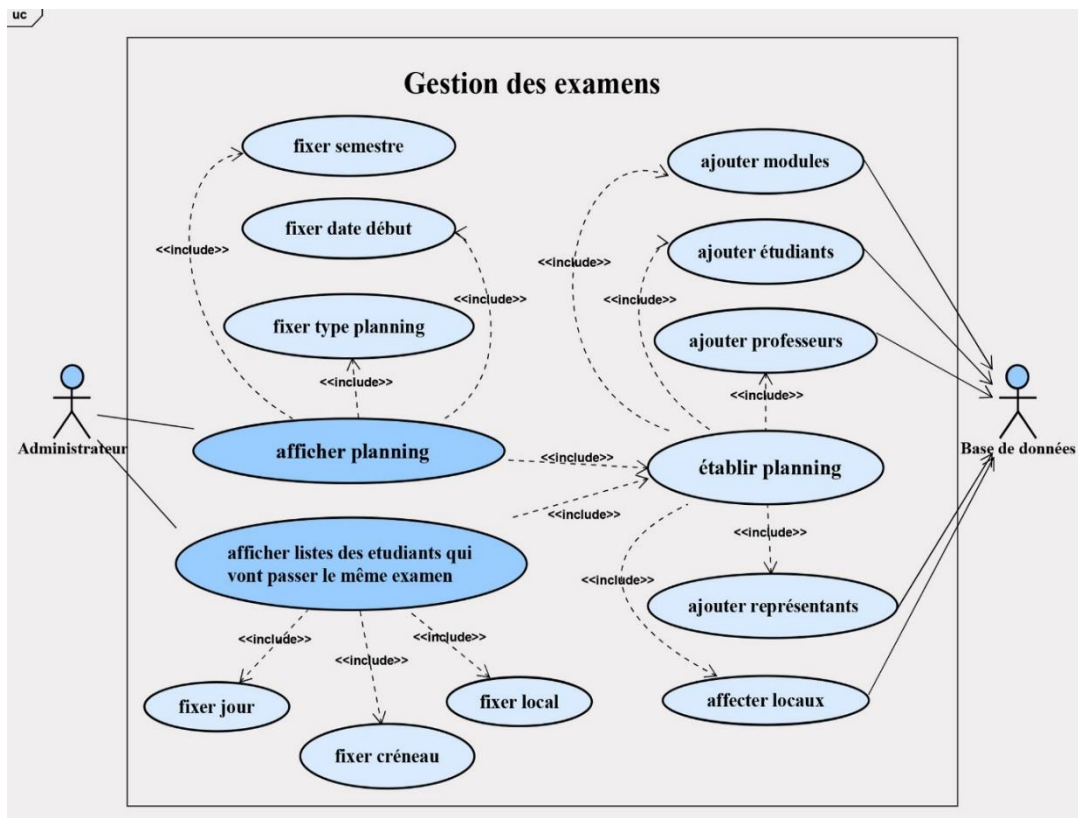


Figure 17 : Diagramme de cas d'utilisation

II.2.2.e. Diagramme de séquence

Dans cette partie nous allons aborder la spécification des besoins détaillé en élaborant pour chaque cas d'utilisation un diagramme de séquence qui illustre les interactions relatives à ce cas entre les acteurs et le système pour un scénario normal.

- Afficher le planning

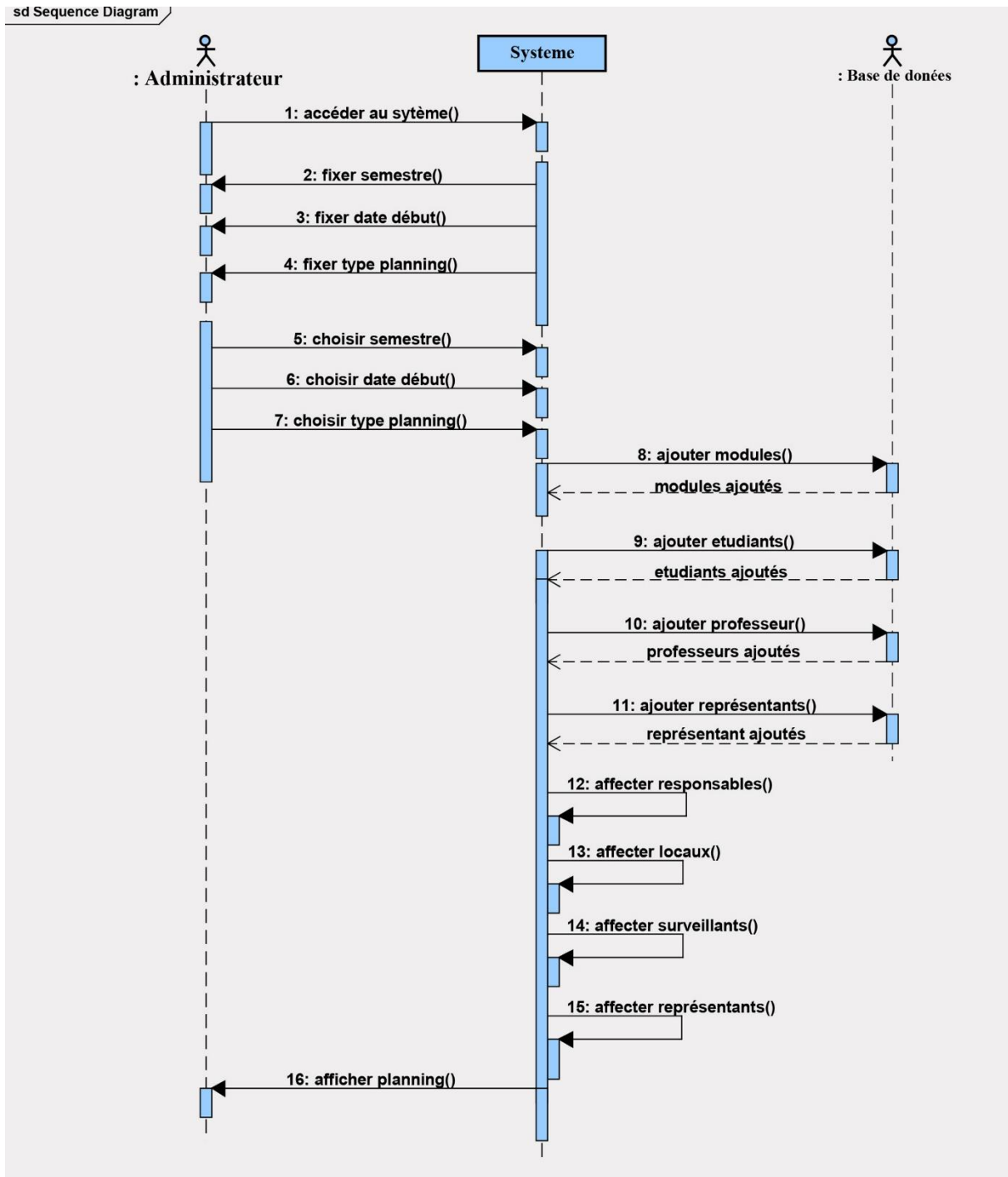


Figure 18 : Diagramme de séquence afficher planning

- Afficher listes des étudiants qui vont passer le même examen

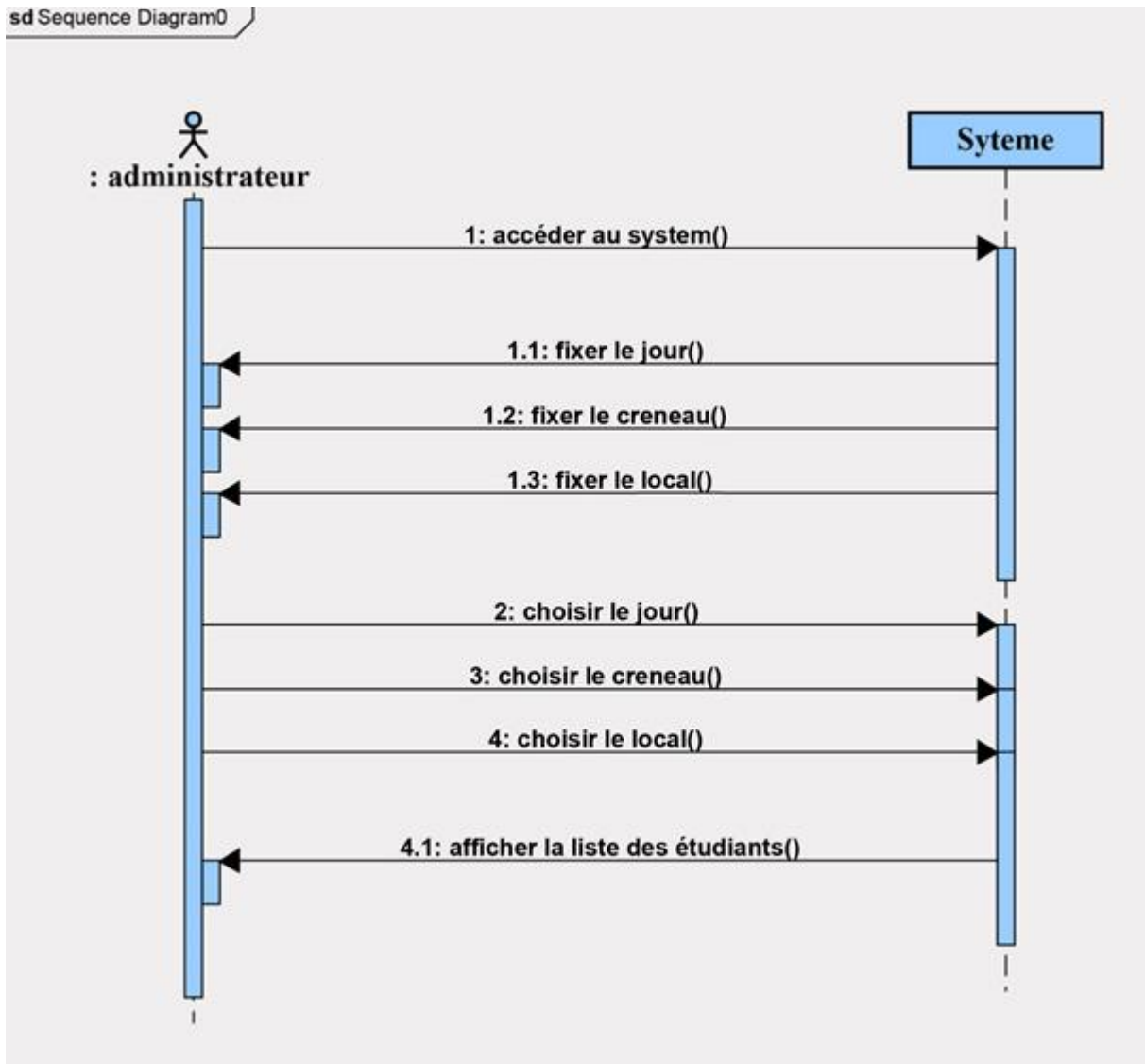


Figure 19 : Diagramme de séquence listes des étudiants

II.2.2.f. Diagramme de classes

Dans cette partie, on va représenter le diagramme de classe pour définir les classes principales du système et les différentes relations entre eux.

- Identification des classes :

- Un planning est composé d'au moins un jour.
- Un jour est composé de quatre créneaux.
- Un créneau est composé d'au moins un examen.
- Un examen apparait dans un seul créneau.
- Un examen aura lieu dans un ou plusieurs locaux.
- Un local ne comporte qu'un examen dans un créneau donné.
- Un examen concerne un et un seul module.
- Un module est concerné par un et un seul examen.

- Un étudiant est inscrit dans au moins un module.
- Un module peut avoir plusieurs étudiants.
- Chaque examen a un professeur responsable au moins.
- Un professeur peut être responsable de plusieurs modules.
- Un professeur surveille dans au moins un local.
- Chaque local a exactement un représentant administratif.
- Chaque local a au moins deux surveillants.

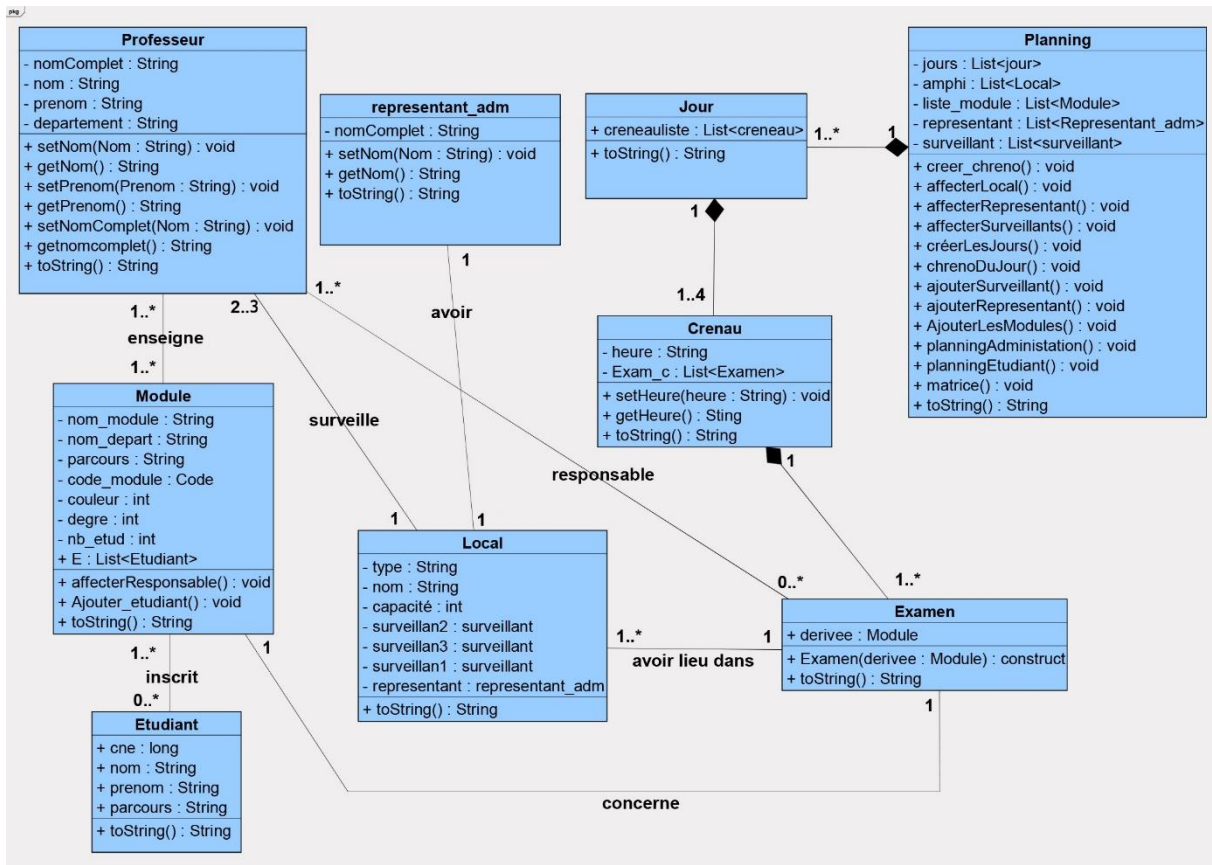


Figure 20 : Diagramme de classes

II.3. Coloration et gestion des examens

Dans cette partie, nous proposons une méthode d'organisation des examens de la FSTE en utilisant la coloration de graphes. Plus précisément nous proposons une modélisation du problème sous forme d'un graphe non orienté, dont les sommets sont les examens. Deux sommets i, j sont adjacents dans le graphe si les examens i et j ont des étudiants en commun ou bien la somme de leurs effectifs dépassent la capacité de l'établissement estimée à 1100 étudiants. Nous attribuons une couleur à chaque sommet de telle sorte que deux sommets adjacents n'aient pas la même couleur.

Donc minimiser le nombre de jours d'organisation des examens, revient à minimiser le nombre des couleurs du graphe.

II.3.1. Graphe d'incompatibilité

La gestion des emplois de temps est concernée par l'attribution d'un ensemble des examens à un ensemble des créneaux horaires, sous un nombre de contraintes qui doivent être satisfaites afin de produire un planning réalisable.

- Les contraintes

- Chaque étudiant ne doit pas avoir deux examens en un même créneau (deux modules ayant un étudiant commun ne peuvent être passés dans le même créneau).
- Chaque créneau ne doit pas avoir deux examens de modules de grande section : La capacité de l'établissement ne doit pas être dépassée à n'importe quelle tranche horaire. On suppose que la capacité des locaux réservés au déroulement des examens peut atteindre 1100 étudiants.

II.3.2. Exemple

On prend par exemple un échantillon de huit examens de la session normale du printemps pour l'année universitaire 2018-2019.

Code	Désignation	Parcours / spécialité
I143	Structure de données en C	DEUST MIP
I616	Programmation web	LST GL
I617	Gestion de projet	LST GL
B244	Biochimie structurale	DEUST BCG
B627	Bio-statistiques	LST PS
B628	Nutrition humaine	LST PS

Tableau 2 : Echantillon de modules

Module i	Module j	Etudiant commun	Capacité dépassée	Adjacence
I143	I616	OUI	-	1
	I617	OUI	-	1
	B244	-	OUI	1
	B627	-	-	0
	B628	-	-	0
I616	I143	OUI	-	1
	I617	OUI	-	1
	B244	-	-	0
	B627	-	-	0
	B628	-	-	0

I617	I616	OUI	-	1
	I143	OUI	-	1
	B244	-	-	0
	B627	-	-	0
	B628	-	-	0
B244	I616	-	-	0
	I617	-	-	0
	I143	-	OUI	1
	B627	-	-	0
	B628	-	-	0
B627	I616	-	-	0
	I617	-	-	0
	I143	-	-	0
	B244	OUI	-	1
	B628	OUI	-	1
B628	I616	-	-	0
	I617	-	-	0
	I143	-	-	0
	B627	OUI	-	1
	B244	OUI	-	1

Tableau 3 : Modules et leurs adjacences

II.3.3.a. Représentation matricielle

Avant de tracer le graphe d'incompatibilité de cet exemple, on s'appuie sur les résultats du tableau ci-dessus, et on modélise ces résultats sous forme d'une matrice carrée qui contient 0 et 1; 1 si i et j sont deux modules qui ont au moins un étudiant en commun ou bien dépassent la capacité de la FSTE et 0 sinon avec $i, j = \{1, \dots, 6\}$.

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

II.3.3.b. Représentation sous forme d'un graphe

On se basant sur la matrice précédente, on va représenter le problème sous forme d'un graphe non orienté dont les sommets sont les modules. Deux examens i et j sont adjacents dans le graphe s'ils ont des étudiants en commun ou ils dépassent la capacité des locaux réservés aux examens.

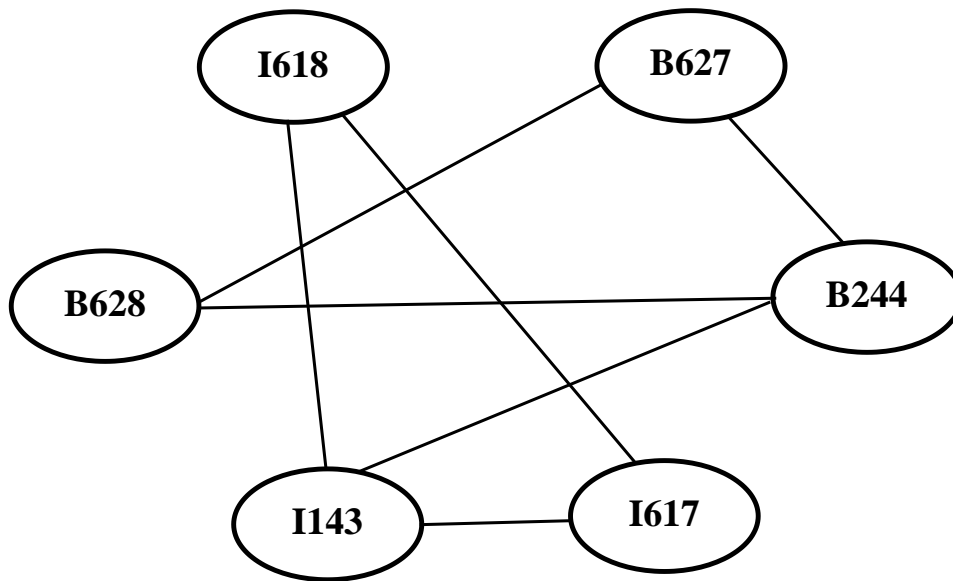


Figure 21 : Graphe d'incompatibilité

II.3.3.c. Coloration

Considérons les couleurs suivantes :



Figure 22 : Couleurs utilisés

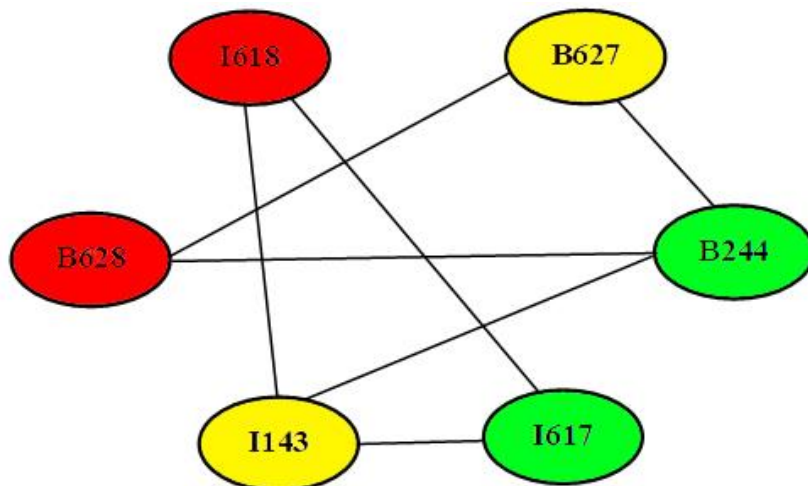


Figure 23 : Résultat de la coloration

Cet exemple académique illustre bien l'idée suivie pour établir un planning des examens répondant aux contraintes déjà mentionnées. On a extrait un échantillon de 6 examens, puis on a utilisé la théorie des graphes pour construire un graphe d'incompatibilité qu'on a coloré manuellement moyennant l'algorithme DSATUR. On a eu comme résultat un graphe coloré avec trois couleurs {I618, B628} en C1, {I143, B627} en C3 et {I617, B244} en C2.

En se basant sur ces résultats, on affirme que deux examens ayant la même couleur puissent avoir lieu dans le même créneau sans problèmes.

Vu le nombre d'examens de cet échantillon, on a pu colorer le graphe manuellement. La coloration manuelle devient très difficile voir même impossible si le graphe est de plus en plus dense. C'est pour cela on a implémenté l'algorithme DSATUR en utilisant le langage de programmation JAVA.

II.4. Conclusion

Ce chapitre était consacré à la conception et la modélisation de notre application à l'aide des diagrammes UML et la coloration des graphes. L'objectif du chapitre suivant sera l'implémentation de tout ce qui est défini dans la conception et la construction d'une interface graphique servante à générer les documents nécessaires afin de mener à bien le déroulement des examens.

Chapitre III : Application à la gestion des examens

III.1. Problématique

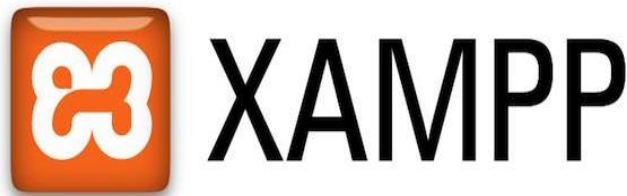
La FSTE doit organiser tous les examens dans le minimum des créneaux horaires possibles, et sous la condition qu'un étudiant ne passe deux épreuves en même temps, et cela en tenant compte des ressources disponibles (locaux, enseignants, administrateurs, ...). Donc l'objectif est de résoudre ce problème via la création d'une interface graphique qui s'appuie sur l'algorithme DSATUR en utilisant JAVA comme langage de programmation en se basant sur la base de données de la faculté.

III.2. Extraction de données

La FSTE dispose d'une base de données pour y stocker les données des étudiants et leurs inscriptions, des modules enseignés, des professeurs et des administrateurs. Notre objectif est d'exploiter cette base de données pour y extraire ces données qui servent à l'implémentation de notre application. Pour cela on aura besoin des moyens qui assurent l'accès à une base de données et son interrogation.

III.2.1. Outils utilisés

III.2.1.a. Serveur XAMPP



XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

Officiellement, XAMPP permet de configurer un serveur de test local.

III.2.1.b. SGBD MySQL



MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

C'est un logiciel libre, open source, développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire.

III.2.2. Connexion à une base de données MySQL

Il existe un driver qui s'ajoute comme extension au projet Java sous forme d'un fichier jar pour assurer la connexion à une base de données MySQL. C'est "mysql_connector_java"[5], le programmeur profite de sa hiérarchie de classes pour établir une connexion à une base de données MySQL.

- Exécution d'une requête SELECT

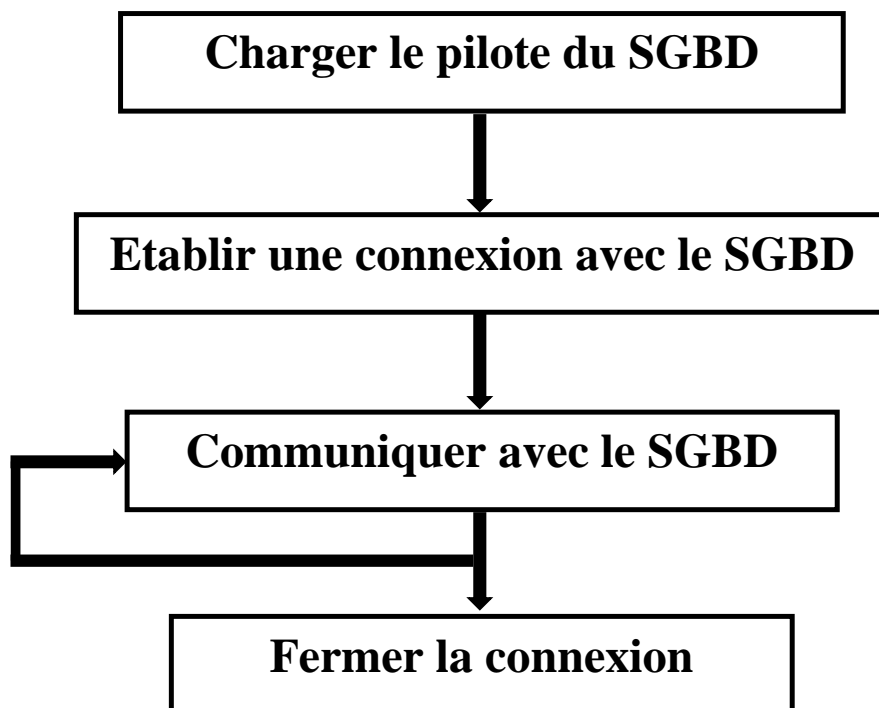


Figure 24 : Principe général pour accéder à une base de données

*** Première étape**

Préciser le type de driver que l'on veut utiliser. Un driver permet de gérer l'accès à un type particulier de SGBD.

Pour MySQL, il faut charger une instance de la classe `com.mysql.jdbc.Driver` pour établir une connexion, c'est pour cela il est nécessaire d'appeler la méthode statique `forName()` de la classe `Class` qui permet de charger dynamiquement une classe dont le nom pleinement qualifié est fourni en paramètre. Elle renvoie une instance de la classe `Class` qui encapsule la classe chargée.

*** Deuxième étape**

Récupérer un objet « Connection » en s'identifiant auprès du SGBD et en précisant la base utilisée.

*** Étapes suivantes**

- A partir de la connexion, créer un « statement » (état) correspondant à une requête de sélection (`SELECT * from Module ...`).
- Exécuter ce statement au niveau du SGBD.
- Parcourir les lignes de la table retourné après l'exécution de la requête.
- Récupérer les valeurs des attributs de la table.
- Fermer le statement.

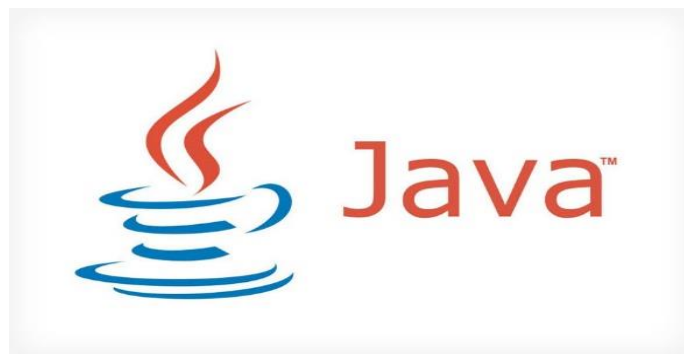
*** Dernière étape**

Se déconnecter de la base en fermant la connexion.

III.3. Noyau de l'application

III.3.1. Outils de développements

III.3.1.a. Langage de programmation : JAVA



La technologie Java définit à la fois un langage de programmation et une plateforme informatique. Créée par l'entreprise Sun Microsystems (souvent juste appelée "Sun"), et reprise depuis par la société Oracle, la technologie Java est indissociable du domaine de l'informatique

et du Web. On la retrouve donc sur les ordinateurs, mais aussi sur les téléphones mobiles, les consoles de jeux, etc.

Le langage de programmation orienté (POO) objet Java permet de développer des applications client-serveur. Les applications développées en Java peuvent fonctionner sur différents systèmes d'exploitation, comme Windows ou Mac OS. Des plugins ajoutés aux navigateurs peuvent toutefois être nécessaire. Java et ces plugins, présents sur de très nombreuses machines, ont connu de nombreuses failles de sécurité, souvent graves et exploités.

III.3.1.b Eclipse IDE



Eclipse IDE est un environnement de développement intégré libre, extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

III.3.2. Représentation des classes et leurs rôles

Dans cette partie, nous présentons les classes du programme, leurs attributs, leurs méthodes et leurs rôles.

- La classe Code

La FSTE suit une règle de codage des modules et chaque code est sous la forme **A x,y,z** avec :

- A : Le nom abrégé du département d'attache du module (M, I, P, B, C, G ou L).
- x : prend la valeur 1 pour un module de DEUST MIP, 2 pour un module de DEUST BCG, 5 pour un module de S5 et 6 pour un module de S6.
- y : pour un module de DEUST, il indique auquel semestre appartient ce module. Pour un module de spécialité, il indique l'option à laquelle appartient ce module.
- z : indique le numéro de module.

Par exemple : I618 c'est un module de S6 spécialité Informatique option 1 et numéro 8. Cette organisation des codes va nous simplifier la planification des modules (suppression, recherche, comparaison, ...).

- La classe Etudiant

Les étudiants inscrits dans un module sont représentés par :

- nom : le nom de l'étudiant.
- prenom : le prénom de l'étudiant.

- parcours : parcours MIP ou BCG.
- cne : le CNE de l'étudiant.

- La classe Module

La classe Module représente tout simplement un module enseigné dans la FSTE. Chaque module possède :

• Attributs :

- nom_module : le nom complet du module.
- nom_depart : le nom du département auquel appartient ce module.
- parcours : BCG ou MIP.
- code_module : de type Code.
- responsable : le professeur responsable du module
- nb_etud : le nombre des étudiants inscrits dans le module.
- degre, indice, couleur : sont des variables utilisées dans la coloration.
- E : une liste des étudiants inscrits dans ce module.

• Méthodes :

- ajouter_etudiant() : cette méthode ajoute les étudiants inscrits dans un module de la base de données à la liste des étudiants E de ce module.
- affecterResponsable() : affecte le professeur responsable d'un module à l'attribut 'responsable' de ce module.

- La classe Professeur

Classe pour la création des professeurs qui enseignent les modules de chaque département. Un professeur est représenté par :

- nomComplet : nom et prénom du professeur.
- departement : le département d'attache du professeur.

- La classe Representant_adm

Classe pour la création des représentants de la scolarité. Un surveillant appartient aussi à l'équipe de surveillance dans un examen et il est représenté dans notre programme par :

- nomComplet : nom et prénom de représentant de la scolarité.

- La classe Local

Classe pour la création des locaux dans lesquels les examens se déroulent. Un local se caractérise par :

- type : représente le type du local (amphi, forum, salle).
- nom : le nom du local.
- capacité : le nombre maximal des étudiants qui peut accueillir.
- liste_E_local : liste des étudiants qui vont passer l'examen dans ce local.
- module : le module qui occupe le local.
- surveillant1, surveillant2, surveillant3 : les professeurs surveillant dans ce local.
- representant : le représentant administrateur affecté à ce local.

- La classe Examen

Les étudiants passent pour chaque module un examen, qui est représenté par :

- responsable : le professeur responsable.
- dérivée : le module concerné par l'examen.

- La classe Créneau

C'est une classe qui représente un créneau horaire pour passer des examens, en évitant des chevauchements.

- Examen_m_c : une liste des modules de même couleur.
- heure : horaire de créneau.

- La classe Jour

Classe qui représente un jour dans le planning, qui est caractérisé par :

- numero_Jour : le numéro du jour.
- creneauxDuJour : une liste des créneaux constituant le jour (d'un à quatre créneaux).

- La classe Planning

Cette classe contient un ensemble des méthodes pour obtenir à la fin un planning sans chevauchements entre les examens. Il est caractérisé par :

• Attributs :

- Jours : la liste des jours constituant le planning.
- Amphi : la liste contenant tous les locaux de la FSTE quel que soit leurs types (forum, amphi, salle).
- Liste_module : la liste des modules.
- Représentant : la liste des représentants de l'administration.
- Surveillants : la liste des professeurs.

• Méthodes :

- AjouterLesModules() : cette méthode remplit la liste 'liste_E_module' par les modules qui ont au moins un étudiants inscrits, d'après la base de données.
- matrice () : cette méthode crée une matrice en se basant sur les contraintes déjà spécifié
- coloration () : attribut une couleur a chaque module
- Créer_chreno() : créer les créneaux de tel sorte que deux modules ayant la même couleur soient regroupés dans un créneaux
- chrenoDuJours () : distribue les créneaux sur les jours
- affecterLocal () : affecte les locaux aux examens
- affecterSurveillants () : affecte les surveillants aux examens

- affecterRepresentants () : affecte les représentant de l'administrations aux examens
- planningEtudiants () : affiche un planning destiné aux étudiants sous forme d'un fichier en format PDF
- planningAdministration() : affiche un planning destiné aux Administrateurs sous forme d'un PDF
- afficherListeDesEtudiants () : affiche une liste des étudiants qui vont passer un examen dans un local dans un jour est horaire bien définie.

III.4. Interface graphique

III.4.1. Moyens utilisés

III.4.1.a Les bibliothèques AWT / SWING

- Généralité sur AWT

- AWT est une interface utilisateur offrant un ensemble de classes et de fonctions pour le développement des interfaces graphiques.
- AWT fut introduite dès java 1.0. Depuis Java 1.2, SWING a très largement remplacé AWT.
- AWT regroupe toutes les fonctionnalités communes à tous les systèmes graphiques pour la manipulation des objets *GUI*.
- AWT fait appel à l'afficheur natif du système local pour dessiner les composantes graphiques.
- AWT n'est pas totalement indépendante de l'OS : apparence différente d'un système à l'autre.
- Avantage : rendement graphique plus rapide.
- Package : java.awt.

- Généralité sur SWING

- C'est une librairie écrite en JAVA.
- Développée par Sun pour remplacer AWT.
- Permet de dessiner des interfaces graphiques complexes.
- Le but étant d'être complètement détaché de l'OS sur lequel l'application sera lancée. Là où AWT était plus lié au système.
- Existe depuis de nombreuses années et bénéficie de nombreuses ressources sur Internet.
- Multiplateforme simple et efficace.
- Package : javax.swing.

- Les principales composantes graphiques

	Composantes	Désignation
Widgets	JLabel	Zone de texte statique
	JButton	Bouton simple
	JCheckbox	Case à cocher
	JRadioButton	Radio bouton
	ButtonGroup	Regroupe un ensemble de radio boutons tel qu'un seul bouton est sélectionné à la fois
	JList	Liste déroulante
	TextField	Zone de texte dynamique
	PasswordField	Zone du mot de passe
Conteneurs	JFrame	Fenêtre simple avec bordures, décorations et boutons
	JPanel	Conteneurs de base qui s'ajoute à JFrame
	JDialog	Une boîte du dialogue

*Tableau 4 : Les principales composantes graphiques***III.4.1.b. La bibliothèque ITEXT**

ITEXT est une bibliothèque Java créé par Bruno LOWAGIE, il s'ajoute comme extension au projet Java à l'aide de driver ITextpdf qui permet de générer, manipuler, et lire les PDFs. Une fois le driver ITextpdf s'est mis dans la bibliothèque du projet comme un externe fichier jar, le programmeur dispose d'une hiérarchie de classes qui lui permet le traitement des PDFs [6].

III.4.2. Apparence générale de l'interface

C'est la première interface qui s'affiche dès qu'on lance notre application.

Figure 25 : Interface de l'application

III.4.3. Affichage du planning

Pour afficher le planning, il est indispensable de :

- Choisir un semestre

Figure 26 : Choix du semestre

Le 1^{er} semestre désigne la session d'automne et le 2^{ème} semestre désigne la session de printemps.

- Choisir une date de début

The screenshot shows a web application window titled 'Gestionnaire des examens'. It has three main sections. The top section, 'Choix du semestre', has two radio buttons: '1^{er} semestre' (selected) and '2^{ème} semestre'. The middle section, 'Choix de la date du début des examens', contains three input fields: 'Jour' (17), 'Mois' (06), and 'Année' (2019). These fields are enclosed in a green rectangular box. To the right of this section, it says 'Nombre total des étudiants : 3236'. The bottom section, 'Affichage du planning', has two radio buttons: 'Planning Enseignants' (selected) and 'Planning Etudiants'. A blue button labeled 'Afficher planning' is on the right. A green arrow points from the date selection area towards the 'Afficher planning' button.

Figure 27 : Choix de la date du début

L'administrateur choisit la date de début des examens.

- Exception si la date est mal choisie

This screenshot shows the same application window as Figure 27, but with an error. In the 'Choix de la date du début des examens' section, the 'Jour' field now contains the value '32' and is highlighted with a red rectangular box. A red arrow points from this box down to a red text message at the bottom of the window that reads 'Date invalide!'. The other elements of the interface remain the same.

Figure 28 : Date invalide

Si la date choisie est incorrecte, un message d'erreur s'affiche en rouge. Par exemple, 32 dans le champ du jour provoque ce message.

- Choisir un type de planning

The screenshot shows the application window with the date fields correctly set (Jour: 17, Mois: 06, Année: 2019). In the 'Affichage du planning' section, the 'Planning Enseignants' radio button is selected and is enclosed in a green rectangular box. A green arrow points from this box towards the 'Afficher planning' button. The 'Date invalide!' message is no longer present.

Figure 29 : Choix du type planning

L'utilisateur a deux choix : planning enseignants et planning étudiants.

Selon le choix de l'utilisateur, l'application génère un document PDF.

- Si l'utilisateur choisit planning enseignants



Planning des examens de la session de Printemps, année universitaire 2018/2019.

Jour : 17-6-2019						
Horaire : 08:00-10:00						
Code	Designation	Responsable	Section	Local	Surveillants	Representant Ad
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Forum	Azzaoui Bouchra Naciri Rajae Lachgar Mohammed	Lali Khalid
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi A	Daoudia Abdelkrim Rhazi Mouhssine Bouda Brahim	Laabibis Nouzha
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi B	Daya Abdelmajid Elmorsli Driss El harrad Imad	Badra Tayeb
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi E	Mezdar Abdessamad Elomary Mohamed Abdou Babaoui Abdelhak	Lakhlifi Mohamed
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi C	El Allam Driss Sakili Dahmmame Elalaoui Abdellah	Chahid Abderrahim
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi D	Arhchoui Hamid Mekkaoui Toufik Bouslami Najia	Ameraouy Sidi Et-Thami
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	L2	Ouadoudi Nadia Senhaji Omar Taouil Rachid	Saada Hassane
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Salle psi	El Ouattassi Omar Filali Zegzouti Younes Raada Abdelazize	Khalfi Ali
G244	Tectonique	Afiri Abdelkhaleq	BCG/S4	A1	Hammouch Zakia Moustabchir Hassane	Madi Sihame
G244	Tectonique	Afiri Abdelkhaleq	BCG/S4	A2	El-Haidani Ahmed Mamouni Abdellah	Tahiri Abderrahman
G244	Tectonique	Afiri Abdelkhaleq	BCG/S4	A3	Touzani Mohammed El Amrani Aumeur	Hanafi Lalla Aziza
G244	Tectonique	Afiri Abdelkhaleq	BCG/S4	A4	Ouanan Mohammed Bouyahia Elmostafa	Amezouar Sara
L222	Langue et communication 2	Nou Mohamed	BCG/S2	A5	Moussaoui Fouad Charroud Ali	Assoussi Abdelhak
L122	Langue et communication 2	Nou Mohamed	MIP/S2	A6	Benamara Ahmed El Harrak Abdelhay	Oultaib Haddou

Figure 30 : Planning enseignants

Ce document contient toutes les informations nécessaires pour le déroulement des examens :

- Jour : chaque jour regroupe un ensemble de créneaux.
- Horaire : laps de temps correspondant à un créneau.
- Code : code du module.
- Désignation : désignation du code du module.
- Responsable : responsable du module.

- Section : parcours (ou spécialité) auquel (à laquelle) le module appartient.
- Local : local où l'examen se déroule (un examen peut occuper plusieurs locaux).
- Surveillants : professeurs qui surveillent.
- Représentant ad : représentant de l'administration.

Le Planning résultant est constitué de 23 créneaux, ce qui donne 6 jours.

- Si l'utilisateur choisit planning étudiants



Planning des examens de la session de Printemps, année universitaire 2018/2019.

Jour : 17-6-2019				
Horaire : 08:00-10:00				
Code	Designation	Responsable	Section	Local
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Forum
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi A
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi B
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi E
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi C
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Amphi D
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	L2
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	MIP/S1	Salle psi
G244	Tectonique	Afiri Abdelkhaleq	BCG/S4	A1
G244	Tectonique	Afiri Abdelkhaleq	BCG/S4	A2
G244	Tectonique	Afiri Abdelkhaleq	BCG/S4	A3
G244	Tectonique	Afiri Abdelkhaleq	BCG/S4	A4
L222	Langue et communication 2	Nou Mohamed	BCG/S2	A5
L122	Langue et communication 2	Nou Mohamed	MIP/S2	A6

Figure 31 : Planning étudiants

Ce document est constitué des éléments formant le planning enseignants sauf les surveillants et les représentants de l'administration. La connaissance de ces derniers n'est pas importante pour l'étudiant.

Le Planning résultant est constitué de 23 créneaux ce qui donne 6 jours.

III.4.4. Affichage des listes des étudiants

Pour afficher les listes des étudiants qui vont passer un examen, l'administrateur fixe le jour et le créneau associé à cet examen dans le planning.

Affichage des listes des étudiants:

Choix du jour:

1
2
3
4
5
6

Choix du créneau:

08:00 - 10:00
10:30 - 12:30
14:30 - 16:30
17:00 - 19:00

Choix du local:

Forum
Amphi A
Amphi B
Amphi E
Amphi C
Amphi D
L2

Figure 32 : Choix du jour et créneau

Après avoir fixé le jour et le créneau, il doit spécifier un local parmi les locaux que cet examen occupe.

Affichage des listes des étudiants:

Choix du jour:

1
2
3
4
5
6

Choix du créneau:

08:00 - 10:00
10:30 - 12:30
14:30 - 16:30
17:00 - 19:00

Choix du local:

Forum
Amphi A
Amphi B
Amphi E
Amphi C
Amphi D
L2

Figure 33 : Choix du local

Au cas où le jour, le créneau ou le local n'est pas choisi, un message d'erreur s'affiche en rouge.

Affichage des listes des étudiants:

Choix du jour:

1
2
3
4
5
6

Choix du créneau:

08:00 - 10:00
10:30 - 12:30
14:30 - 16:30
17:00 - 19:00

Choix du local:

Forum
Amphi A
Amphi B
Amphi E
Amphi C
Amphi D
L2

**Verifiez si vous avez
Selectionné un
jour,un créneau et un
local.**

Figure 34 : Jour, créneau ou local non choisi

Ici, l'utilisateur n'a choisi aucun local.

- Exemple

Le module M111 apparait dans le premier jour et le premier créneau (08 :00 -10 :00) et occupe plusieurs locaux. On veut afficher la liste des étudiants qui passent cet examen dans le forum.

Affichage des listes des étudiants:

Choix du jour:

1
2
3
4
5
6

Choix du créneau:

08:00 - 10:00
10:30 - 12:30
14:30 - 16:30
17:00 - 19:00

Choix du local:

Forum
Amphi A
Amphi B
Amphi E
Amphi C
Amphi D

Figure 35 : Sélection du jour, créneau et local

Ainsi, la liste correspondante au choix ci-dessus est la suivante :



Code Module	Désignation	Responsable	Local	Horaire
M111	Analyse 1: Fonction d'une variable réelle	Sidi Ammi Moulay Rchid	Forum	08:00-10:00

CNE	Nom et Prénom	Numero de l'Examen
1513008394	Aabi Oumaima	1
1210915842	Aalla Yassine	2
1311742108	Aazbi Abderrahim	3
1513792662	Abad Kamal	4
1311933233	Abahani Mohamed	5
1513742642	Abboubi Mohammed	6
1210852333	Abdeljalil Salah_Eddine	7
1513935692	Abibi Zakariya	8
1513936578	Abkari Achraf	9
1513935189	Aboubekr Marouane	10
1513851228	Achcha Abdelali	11
1210389001	Addali Asmaa	12

Figure 36 : Liste des étudiants

III.5. Interprétation des résultats

Pour un ensemble de 3236 étudiants inscrits aux différents modules dans les deux parcours MIP et BCG incluant toutes les filières de la FSTE, on a pu planifier les examens pour tous ces étudiants en minimisant la durée, maximisant l'exploitation des ressources disponibles et en évitant les chevauchements.

Ce résultat est obtenu en utilisant l'algorithme DSATUR, une coloration est appliquée à un ensemble de 97 sommets qui représentent 97 modules. On a obtenu 23 couleurs, ce nombre représente le nombre de créneaux qu'il fallait créer pour une 'bonne' distribution des examens.

Une journée est équivalente à quatre créneaux (2 heures par examen), donc la durée nécessaire pour passer tous les examens est 6 jours (quatre créneaux pour les cinq premiers jours et trois pour le sixième).

On remarque qu'un créneau peut avoir un mélange des examens enseignés dans la FSTE quel que soit le parcours ou la spécialité auquel ou à laquelle ils appartiennent, à condition que la somme de leurs effectifs ne dépasse pas la capacité maximale de l'établissement (estimée à 1100 étudiants) et qu'ils n'aient pas un étudiant commun.

Chaque local a une capacité estimée, et les étudiants d'un module sont affectés aux locaux de tel sorte que :

- La capacité d'un local ne soit pas dépassée.
- Un local ne comporte que les étudiants qui passent le même examen, même si la capacité du local n'est pas atteinte.

Les surveillants et les représentants de l'administration sont attachés aux locaux d'une façon circulaire, c'est-à-dire qu'un surveillant ou un représentant n'aura pas une deuxième séance de surveillance que si tous les autres aient fait leurs premières séances. Autrement dit, la répartition des surveillances pour les enseignants et aussi pour le corps administratif est faite d'une manière équitable.

Pour tous les locaux, un représentant de l'administration est affecté et deux surveillants sont associés si la capacité du local est inférieure à 60, et trois sinon.

On ne peut pas trouver un professeur responsable d'un module appartenant à un créneau dans la liste des surveillants de même créneau.

Pour récapituler, il s'agit d'une application qui gère la planification des examens de la FSTE pour 3236 étudiants inscrits. Cela a donné lieu à :

- 6 jours pour le déroulement de tous les examens.
- Au maximum on a un créneau qui contiennent 10 examens dans lequel le forum, les amphithéâtres, les salles du bloc A et une salle du bloc B sont tous occupés.
- Au minimum on a un créneau contenant un examen qui occupe le forum et tous les amphithéâtres.

- Au maximum on aura besoin de 89 surveillants (incluant les représentants de l'administration) par créneau.
- Au minimum on aura besoin de 24 surveillants (incluant les représentants de l'administration) par créneau.
- La durée est minimisée et les ressources (les professeurs, les administrateurs, les locaux) sont exploitées au maximum.
- Ces résultats ne sont pas statiques, ils varient avec l'effectif total des étudiants.

Conclusion générale

La gestion des examens est un problème majeur pour les institutions éducatives et universitaires. Le défi est de garantir les bonnes conditions d'évaluation des étudiants et de réduire au maximum la durée. Les méthodes manuelles traditionnelles ont montré leurs limites dans la gestion de grands flux d'informations notamment lorsque le nombre d'étudiants est important. De ce fait, l'intérêt d'automatiser le processus de gestion des examens devient évident. Cette problématique a été le point de départ de notre projet de fin d'étude. On a fixé comme objectif de minimiser la durée de déroulement des examens sans produire des chevauchements.

Pour atteindre notre objectif, notre application a été basée sur la coloration DSATUR. Les résultats étaient dans les normes et la durée obtenue était inférieure à la durée donnée par la méthode traditionnelle dans le respect de toutes les contraintes, surtout celle de la capacité des locaux de la faculté et la contrainte qu'aucun étudiant ne passe deux examens en même temps.

L'application créée offre aux administrateurs de la FSTE des services relatifs à la gestion des examens, elle leur permet d'avoir l'ensemble des créneaux utilisés pour la planification des examens d'une session donnée distribués sur un ensemble de jours dont chacun est repéré par une date. Chaque créneau contient l'ensemble des examens qui peuvent être passés en même temps sans dépasser la capacité de la faculté. Aussi elle leur permet de consulter pour un examen sa liste des étudiants et son équipe de surveillance.

Au-delà de l'intérêt et du bénéfice que pourraient avoir ce projet pour les institutions éducatives et universitaires, sa réalisation a été, pour nous, une véritable expérience pédagogique et de recherche. En effet, il nous a permis de vivre pratiquement toutes les étapes de développement d'applications.

Suite à ce travail, nous comptons faire des extensions à notre application. En effet, elle s'intéresse seulement à la phase d'avant les examens, on voudrait étendre ses fonctionnalités pour inclure la phase d'après les examens, c'est-à-dire d'atténuer les problèmes qui se posent lors de cette phase, notamment lors de l'affichage des notes, de la correction et de la consultation des copies corrigées. Concernant l'affichage des notes, le système offrirait à tous les étudiants la possibilité de consulter leurs notes à tout moment, évitant ainsi les désagréments de la méthode classique où l'affichage n'est pas facilement accessible (bousculade, liste d'affichage illisibles, ...etc.). Contrairement à la pratique traditionnelle des consultations effectuées au bureau de l'enseignant qui engendre beaucoup de stress pour l'enseignant et l'étudiant, l'idée est de donner à chaque étudiant les moyens nécessaires pour consulter sa copie et poster des réclamations éventuelles. D'autre part, avec ce système, l'enseignant aurait suffisamment de temps pour traiter chaque réclamation.

Bibliographie

- [1] Claude Berge, Théorie des graphes et ses applications, Paris, Dunod, 275 p, 1958.
- [2] Claude Berge, *Hypergraphes. Combinatoires des ensembles finis*, Bordas, 1987.
- [3] D. Brélaz, "New methods to color the vertices of a graph," *Communications of the ACM*, vol. 22, no. 4, pp. 251-256, 1979.
- [4] R.M.R Lewis, A Guide to Graph Colouring, 2015.

Webographie

- [5] mysql_connector_java pilote récupéré sur :
<https://dev.mysql.com/downloads/connector/j/5.1.html>
- [6] ITEXT récupéré sur : <https://itextpdf.com/fr>